MT6516 SM/GPRS/EDGE Application Processor

Data Sheet

Version:	1.02
Release date:	2009-05-05

© 2009 MediaTek Inc.
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

Revision History

Revision	Date	Comments
1.00	Feb 24, 2009	First release
1.01	April 1, 2009	Minor format change Updated pin definitions: W31, W33, AA37, AB32
1.02	May 5, 2009	Removed one debug pin. Updated auxiliary ADC unit section Modified MSDC CLKSRC description of MSDC_CFG Added IrDA framer section Added MPEG-4/H.263 Video CODEC

© 2009 MediaTek Inc.

TABLE OF CONTENTS

Revision History **2**

Preface **6**

1 Product Description **7**

1.1 Pin Outs 7
1.2 Top Marking Definition 9
1.3 DC Characteristics 9
1.4 Pin Description 10
1.5 Power Description 20
1.6 Ordering information 20

2 Application Micro-Controller Unit Subsystem **20**

2.1 Processor Core 20
2.2 Memory Management 20
2.3 Bus System 20
2.4 UUID 20
2.5 Interrupt Controller 20
2.6 Direct Memory Access 20
2.7 AP CONFIG Register 20
2.8 APMCUSYS CONFIG Register 20
2.9 AP eXtended GPT 20
2.10 Auxiliary ADC Unit 20
2.11 CoreSight 20
2.12 CPU-CPU interface (CCIF) 20
2.13 EFUSE Controller (efusec) 20
2.14 External Memory Interface 20
2.15 General Purpose Inputs/Outputs 20
2.16 General Purpose Timer (AP) 20
2.17 GRAPH1SYS Clock Management Register 20
2.18 GRAPH2SYS Clock Management Register 20
2.19 HDQ/1-wire 20
2.20 I2C / SCCB Controller 20
2.21 IrDA Framer 20
2.22 Keypad Scanner 20
2.23 Memory Stick and SD Memory Card Controller 20
2.24 NAND FLASH Interface 20
2.25 NAND FLASH ECC 20
2.26 Reset Generation Unit (APRGU) 20
2.27 SIM Interface 20
2.28 Slow Clocking Unit for AP side 20
2.29 UART 20

 © 2009 MediaTek Inc.

2.30 USB 2.0 High-Speed Dual-Role Controller........20

3 Modem Micro-Controller Unit Subsystem........20

3.1 Processor Core........20
3.2 Memory Management........20
3.3 Bus System........20
3.4 Interrupt Controller........20
3.5 Direct Memory Access........20
3.6 General Purpose Inputs/Outputs........20
3.7 General Purpose Timer (MD)........20
3.8 L1 Cache controller........20
3.9 MPU........20
3.10 Log Accelerator........20
3.11 MD CONFIG Register........20
3.12 MDMCUSYS CONFIG Register........20
3.13 Reset Generation Unit (MDRGU)........20

4 2.75G Modem Subsystem........20

4.1 Automatic Frequency Control (AFC) Unit........20
4.2 Automatic Power Control (APC) Unit........20
4.3 Baseband Front End........20
4.4 Baseband Parallel Interface........20
4.5 Baseband Serial Interface........20
4.6 CSD Accelerator........20
4.7 Divider........20
4.8 FCS Codec........20
4.9 GPRS Cipher Unit........20
4.10 MD2GSYS CONFIG Register........20
4.11 Timing Generator........20
4.12 Voice Front-End........20

5 Multimedia Subsystem........20

5.1 2D acceleration........20
5.2 Audio SRC Mixer........20
5.3 Backlight Scaling........20
5.4 Camera Interface........20
5.5 Camera Interface........20
5.6 Capture Resize........20
5.7 CEVASYS Subsystem........20
5.8 Display Pixel Interface Controller........20
5.9 Drop Resize........20
5.10 Display Serial Interface Controller........20
5.11 Graphics Memory Interface........20
5.12 GMC Fake Engine........20

 © 2009 MediaTek Inc.

5.13 GRAPH1SYS CONFIG Register 20
5.14 GRAPH2SYS CONFIG Register 20
5.15 H.264 Decoder 20
5.16 Image DMA 20
5.17 Image Processor 20
5.18 JPEG Decoder 20
5.19 JPEG Encoder 20
5.20 LCD Interface 20
5.21 M3D 20
5.22 MPEG-4 Deblocking Filters 20
5.23 MPEG-4/H.263 Video CODEC 20
5.24 Post Resize 20
5.25 SPI Interface Controller 20
5.26 TV Controller 20
5.27 TV encoder 20
5.28 Wavetable Synthesizer 20

6 Clock, Mixed Subsystem 20

6.1 Analog Front-end & Analog Blocks 20
6.2 Clocks 20
6.3 Pulse-Width Modulation Outputs 20
6.4 Real Time Clock 20

 © 2009 MediaTek Inc.

Preface

Acronym for Register Type

R/W	Capable of both read and write access
RO	Read only
RC	Read only. After reading the register bank, each bit which is HIGH(1) will be cleared to LOW(0) automatically.
WO	Write only
W1S	Write only. When writing data bits to register bank, each bit which is HIGH(1) will cause the corresponding bit to be set to 1. Data bits which are LOW(0) has no effect on the corresponding bit.
W1C	Write only. When writing data bits to register bank, each bit which is HIGH(1) will cause the corresponding bit to be cleared to 0. Data bits which are LOW(0) has no effect on the corresponding bit.

© 2009 MediaTek Inc.

1 Product Description

1.1 Pin Outs

One type of package for this product, TFBGA 15mm*15mm, 564-ball, 0.378mm pitch (0.53457mm stagger) Package, is offered.

Pin outs and the top view are illustrated in **Figure 1** for this package. Outline and dimension of package is illustrated in **Figure 2**, while the definition of package is shown in **Table 1**.

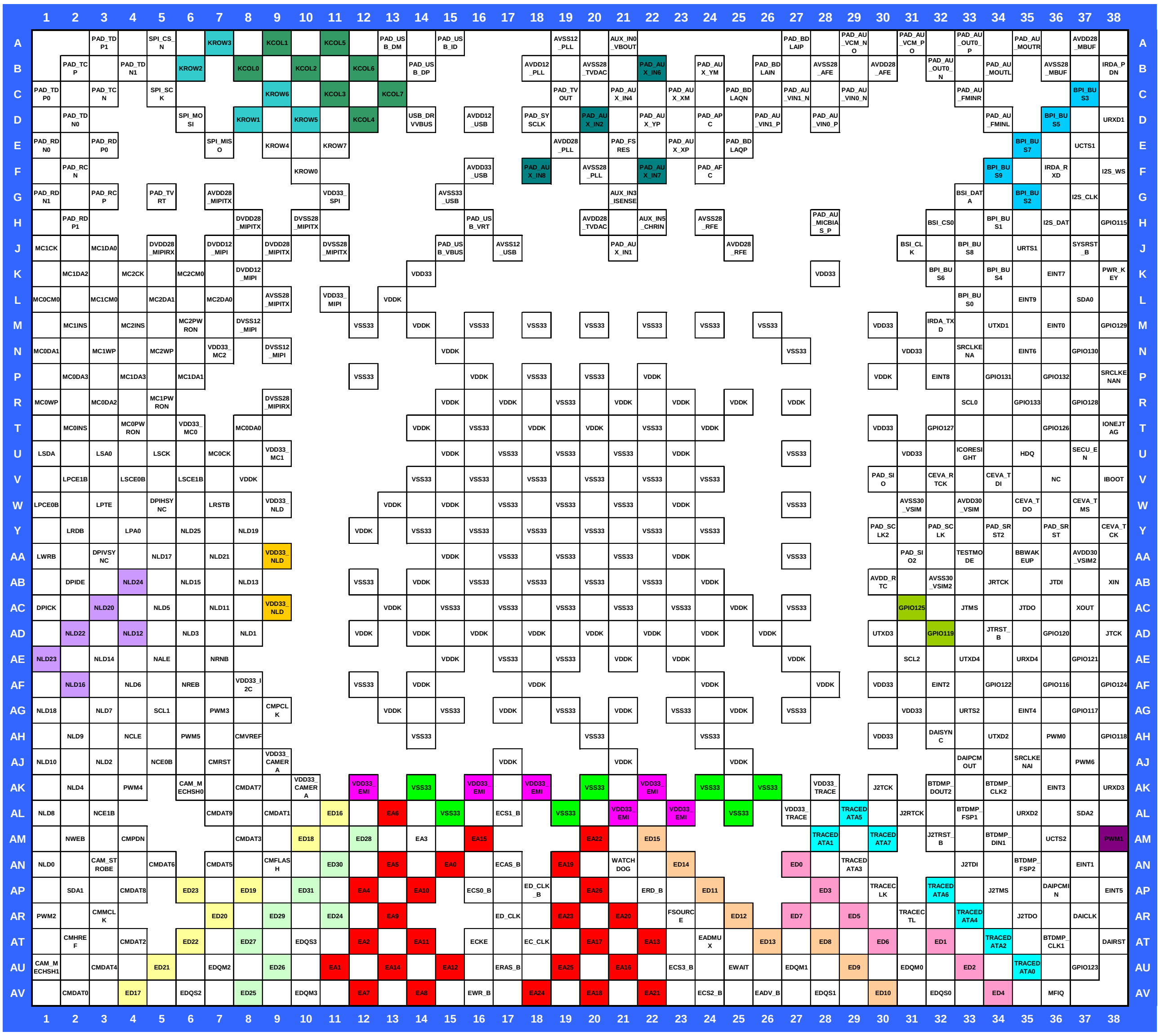
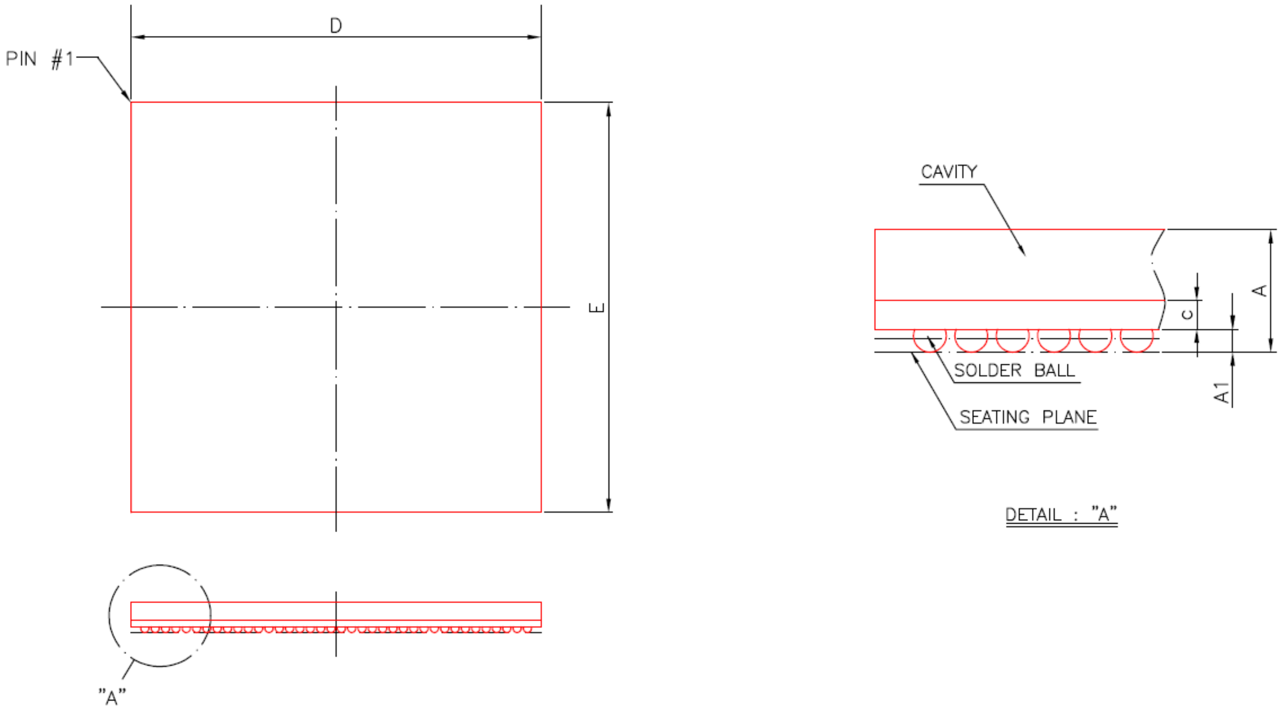


Figure 1 Top View of MT6516 TFBGA 15mm*15mm, 564-ball, 0.378 mm Pitch Package*

MT6516 Top View (Balls Facing Down)



MT6516 Bottom View

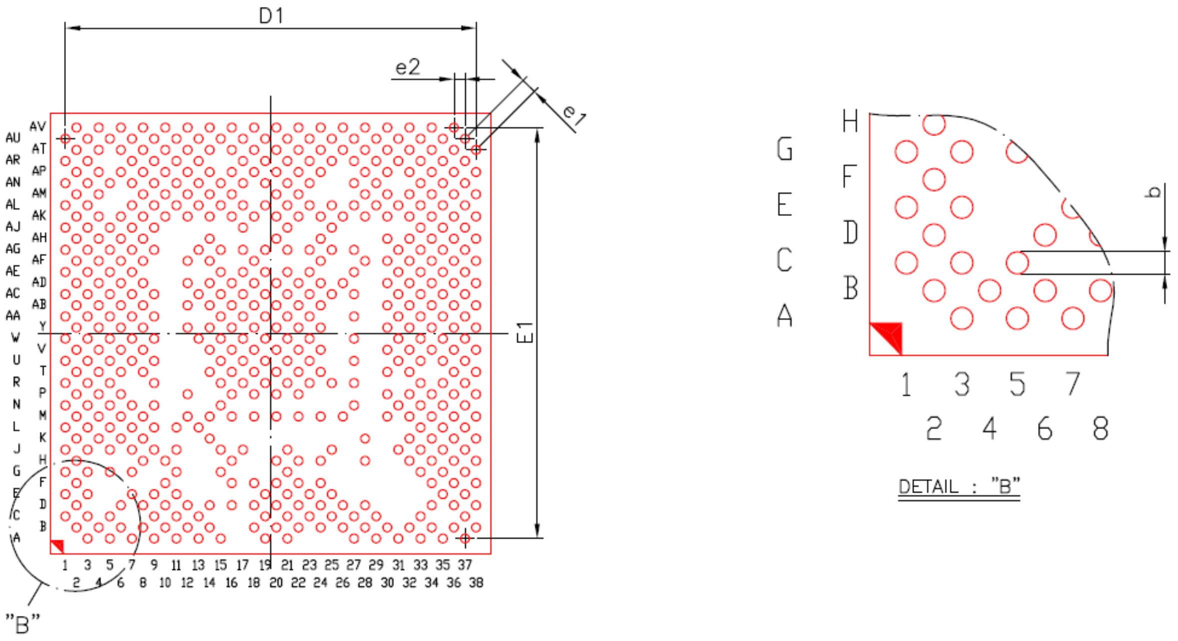
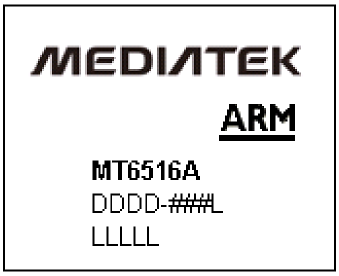


*Figure 2 Outlines and Dimension of TFBGA 15mm*15mm, 564-ball, 0.378 mm Pitch Package*

	Package Size		Edge Ball Center to Center		Ball Pitch	Ball Dia.	Package Thk.	Stand Off	Substrate Thk.
Symbol	D	E	D1	E1	e1/e2	b	A (Max.)	A1	C
Dimension in mm	15	15	13.99	13.99	0.535/0.378	0.3	1.2	0.21	0.26

*Table 1 Definition of TFBGA 15mm*15mm, 564-ball, 0.378 mm Pitch Package (Unit: mm)*

1.2 Top Marking Definition



MT6516A	Part Number
DDDD	Date Code
###	Subcontractor Code
LLLLL	Lot Number

Figure 3 MT6516 Top Marking

1.3 DC Characteristics

1.3.1 Absolute Maximum Ratings

Prolonged exposure to absolute maximum ratings may reduce device reliability. Functional operation at these maximum ratings is not implied.

Item	Symbol	Min	Max	Unit
IO power supply	VDD33	-0.3	VDD33+0.3	V
I/O input voltage	VDD33I	-0.3	VDD33+0.3	V
Operating temperature	Topr	-20	80	Celsius
Storage temperature	Tstg	-55	125	Celsius

1.4 Pin Description

Table 2 Pin Descriptions (Bolded types are functions at reset.)

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			UART Interface				
D38	**URXD1**	IO	UART 1 receive data				
M34	**UTXD1**	IO	UART 1 transmit data				
E37	**UCTS1**	IO	UART 1 clear to send				
J35	**URTS1**	IO	UART 1 request to send				
AL35	**URXD2**	IO	UART 2 receive data				
AH34	**UTXD2**	IO	UART 2 transmit data				
AM36	**UCTS2**	IO	UART 2 clear to send	**GPIO109**	I:UCTS2		
AG33	**URTS2**	IO	UART 2 request to send	**GPIO108**	O:URTS2		
AK38	**URXD3**	IO	UART 3 receive data	**GPIO67**	I:URXD3	I:UCTS4	
AD30	**UTXD3**	IO	UART 3 transmit data	**GPIO68**	O:UTXD3	O:URTS4	
AE35	**URXD4**	IO	UART 4 receive data	**GPIO69**	I:URXD4	I:UCTS3	
AE33	**UTXD4**	IO	UART 4 transmit data	**GPIO70**	O:UTXD4	O:URTS3	
			Dedicated GPIO Interface				
H38	**GPIO115**	IO	General purpose input/output 115	**GPIO115**	O: CLKM0		
AF36	**GPIO116**	IO	General purpose input/output 116	**GPIO116**	O: CLKM1		
AG37	**GPIO117**	IO	General purpose input/output 117	**GPIO117**	O: CLKM2		
AH38	**GPIO118**	IO	General purpose input/output 118	**GPIO118**	O: CLKM3		
AD32	**GPIO119**	IO	General purpose input/output 119	**GPIO119**	O: CLKM4		
AD36	**GPIO120**	IO	General purpose input/output 120	**GPIO120**			
AE37	**GPIO121**	IO	General purpose input/output 121	**GPIO121**			
AF34	**GPIO122**	IO	General purpose input/output 122	**GPIO122**			
AU37	**GPIO123**	IO	General purpose	**GPIO123**			

 © 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			input/output 123				
AF38	**GPIO124**	IO	General purpose input/output 124	**GPIO124**			
AC31	**GPIO125**	IO	General purpose input/output 125	**GPIO125**			
T36	**GPIO126**	IO	General purpose input/output 126	**GPIO126**			
T32	**GPIO127**	IO	General purpose input/output 127	**GPIO127**			
R37	**GPIO128**	IO	General purpose input/output 128	**GPIO128**			
M38	**GPIO129**	IO	General purpose input/output 129	**GPIO129**			I:UCTS4
N37	**GPIO130**	IO	General purpose input/output 130	**GPIO130**			O:URTS4
P34	**GPIO131**	IO	General purpose input/output 131	**GPIO131**			I:UCTS3
P36	**GPIO132**	IO	General purpose input/output 132	**GPIO132**			O:URTS3
R35	**GPIO133**	IO	General purpose input/output 133	**GPIO133**			
			IrDA Interface				
F36	**IRDA_RXD**	IO	IrDA receive data	**GPIO84**	I:IRDA_RXD	CEVA_GPIO0	I:MFIQ
M32	**IRDA_TXD**	IO	IrDA transmit data	**GPIO85**	O:IRDA_TXD	CEVA_GPIO1	
B38	**IRDA_PDN**	IO	IrDA Power Down Control	**GPIO86**	O:IRDA_PDN	CEVA_GPIO2	
			SIM Card Interface				
Y36	**PAD_SRST**	IO	SIM card 1 reset output				
Y32	**PAD_SCLK**	IO	SIM card 1 clock output				
V30	**PAD_SIO**	IO	SIM card 1 data input/output				
Y34	**PAD_SRST2**	IO	SIM card 2 reset output				
Y30	**PAD_SCLK2**	IO	SIM card 2 clock output				
AA31	**PAD_SIO2**	IO	SIM card 2 data input/output				
			Keypad Interface				
C13	**KCOL7**	IO	Keypad column 7	**GPIO71**	I:KCOL7	O:CLKM4	CEVA_GPIO14
B12	**KCOL6**	IO	Keypad column 6	**GPIO72**	I:KCOL6	O:CLKM5	CEVA_GPIO15
A11	**KCOL5**	IO	Keypad column 5	**GPIO28**	I:KCOL5	I:D2ICK	O:usb_probe_out[4]
D12	**KCOL4**	IO	Keypad column 4	**GPIO91**	I:KCOL4	CEVA_GPIO7	I:EINT14
C11	**KCOL3**	IO	Keypad column 3	**GPIO90**	I:KCOL3	CEVA_GPIO6	I:EINT13
B10	**KCOL2**	IO	Keypad column 2	**GPIO89**	I:KCOL2	CEVA_GPIO5	I:EINT12

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
A9	**KCOL1**	IO	Keypad column 1	**GPIO88**	I:KCOL1	CEVA_GPIO4	I:EINT11
B8	**KCOL0**	IO	Keypad column 0	**GPIO87**	I:KCOL0	CEVA_GPIO3	I:EINT10
E11	**KROW7**	IO	Keypad row 7	**GPIO73**	KROW7	O:CLKM6	CEVA_GPIO16
C9	**KROW6**	IO	Keypad row 6	**GPIO74**	KROW6	O:CLKM7	CEVA_GPIO17
D10	**KROW5**	IO	Keypad row 5	**GPIO23**	KROW5	O:DSP2_GPO1	
E9	**KROW4**	IO	Keypad row 4	**GPIO114**	KROW4		
A7	**KROW3**	IO	Keypad row 3	**GPIO113**	KROW3		
B6	**KROW2**	IO	Keypad row 2	**GPIO112**	KROW2		
D8	**KROW1**	IO	Keypad row 1	**GPIO111**	KROW1		
F10	**KROW0**	IO	Keypad row 0	**GPIO110**	KROW0		
K38	**PWR_KEY**	IO	Dedicated KEY for power detection				
			JTAG Port				
AD34	**JTRST_B**	I	JTAG test port reset input	**GPIO48**	I:JTRST_B	O:CLKM6	
AD38	**JTCK**	I	JTAG test port clock input	**GPIO49**	I:JTCK	O:CLKM7	
AB36	**JTDI**	I	JTAG test port data input	**GPIO50**	I:JTDI		
AC33	**JTMS**	I	JTAG test port mode switch	**GPIO51**	I:JTMS		
AC35	**JTDO**	IO	JTAG test port data output	**GPIO52**	JTDO		

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AB34	**JRTCK**	IO	JTAG test port returned clock output	**GPIO53**	JRTCK		
AM32	**J2TRST_B**	I	JTAG test port reset input	**GPIO100**	I:J2TRST_B		O:TRACEDATA8
AK30	**J2TCK**	I	JTAG test port clock input	**GPIO101**	I:J2TCK		O:TRACEDATA9
AN33	**J2TDI**	I	JTAG test port data input	**GPIO102**	I:J2TDI		O:TRACEDATA10
AP34	**J2TMS**	I	JTAG test port mode switch	**GPIO103**	I:J2TMS		O:TRACEDATA11
AR35	**J2TDO**	IO	JTAG test port data output	**GPIO104**	J2TDO		O:TRACEDATA12
AL31	**J2RTCK**	IO	JTAG test port returned clock output	**GPIO105**	J2RTCK		O:TRACEDATA13
			Miscellaneous				
J37	**SYSRST_B**	I	System reset input active low				
AN21	**WATCHDOG**	IO	Watchdog reset output				
N33	**SRCLKENA**	IO	External TCXO enable output active high	**GPIO56**	O:SRCLKENA	I:EINT18	
P38	**SRCLKENAN**	IO	External TCXO enable output active low	**GPIO57**	O:SRCLKENAN	I:EINT19	
AJ35	**SRCLKENAI**	IO	External TCXO enable input	**GPIO58**	O:SRCLKENAI	I:EINT20	
V38	**IBOOT**	I	Boot Device Configuration Input				
AT24	**EADMUX**	IO	NOR/PSRAM A/D Mux bus selection	**GPIO1**	I:EADMUX	O:CLKM1	I:EINT15
V36	**NC**	I	No connection				
U37	**SECU_EN**	I	Security Configuration Input				
U35	**HDQ**	IO	HDQ	**GPIO107**	HDQ		
F38	**I2S_WS**	IO	I2S_WS	**GPIO25**	I2S_WS	I:D1ICK	O:usb_probe_out[7]
H36	**I2S_DAT**	IO	I2S data	**GPIO27**	I2S_DAT	I:D1IMS	O:usb_probe_out[5]
G37	**I2S_CLK**	IO	I2S clock	**GPIO26**	I2S_CLK	D1ID	O:usb_probe_out[6]

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
U33	**ICORESIGHT**	I	Coresight Configuration Input				
T38	**IONEJTAG**	I	Serial JTAG Enable				
D14	**USB_DRVVBUS**	IO	USB OTG Host Mode 5V Charge Pump Enable	**GPIO32**	USB_DRVVBUS		
			External Interrupt				
M36	**EINT0**	IO	External interrupt 0	**GPIO59**	I:EINT0		CEVA_GPIO18
AN37	**EINT1**	IO	External interrupt 1	**GPIO60**	I:EINT1		CEVA_GPIO19
AF32	**EINT2**	IO	External interrupt 2	**GPIO61**	I:EINT2		CEVA_GPIO20
AK36	**EINT3**	IO	External interrupt 3	**GPIO62**	I:EINT3		CEVA_GPIO21
AG35	**EINT4**	IO	External interrupt 4	**GPIO63**	I:EINT4		CEVA_GPIO22
AP38	**EINT5**	IO	External interrupt 5	**GPIO64**	I:EINT5		CEVA_GPIO23
N35	**EINT6**	IO	External interrupt 6	**GPIO65**	I:EINT6		CEVA_GPIO24
K36	**EINT7**	IO	External interrupt 7	**GPIO66**	I:EINT7		CEVA_GPIO25
P32	**EINT8**	IO	External interrupt 8	**GPIO21**	I:EINT8	O:DSP_GPO1	O:TBRXEN
L35	**EINT9**	IO	External interrupt 9	**GPIO22**	I:EINT9	O:DSP_GPO0	O:TBRXFS
AV36	**MFIQ**	IO	Interrupt to MCU				
			SPI Interface				
A5	**SPI_CS_N**	IO	SPI chip select	**GPIO80**	O:SPI_CS_N	I:IRDA_RXD	O:BSI_CS1
C5	**SPI_SCK**	IO	SPI serial clock	**GPIO81**	O:SPI_SCK	O:IRDA_TXD	
D6	**SPI_MOSI**	IO	SPI master output (slave input)	**GPIO82**	O:SPI_MOSI	O:IRDA_PDN	MC2DA2
E7	**SPI_MISO**	IO	SPI master input (slave output	**GPIO83**	I:SPI_MISO	I:MIRQ	MC2DA3
			Digital Audio Interface				
AR37	**DAICLK**	IO	DAI clock output	**GPIO75**	O:DAICLK		CEVA_GPIO26
AJ33	**DAIPCMOUT**	IO	DAI pcm data out	**GPIO76**	O:DAIPCMOUT		CEVA_GPIO27
AP36	**DAIPCMIN**	IO	DAI pcm data input	**GPIO77**	I:DAIPCMIN		CEVA_GPIO28
AT38	**DAIRST**	IO	DAI reset signal input	**GPIO78**	I:DAIRST	O:CLKM5	CEVA_GPIO29
AH32	**DAISYNC**	IO	DAI frame synchronization	**GPIO79**	O:DAISYNC		CEVA_GPIO30

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			signal output				
			RF Parallel Control Unit				
L33	**BPI_BUS0**	IO	RF hard-wire control bus 0				
H34	**BPI_BUS1**	IO	RF hard-wire control bus 1				
G35	**BPI_BUS2**	IO	RF hard-wire control bus 2				
C37	**BPI_BUS3**	IO	RF hard-wire control bus 3				
K34	**BPI_BUS4**	IO	RF hard-wire control bus 4				
D36	**BPI_BUS5**	IO	RF hard-wire control bus 5				
K32	**BPI_BUS6**	IO	RF hard-wire control bus 6	**MD_GPIO0**	O:BPI_BUS6		
E35	**BPI_BUS7**	IO	RF hard-wire control bus 7	**MD_GPIO1**	O:BPI_BUS7		
J33	**BPI_BUS8**	IO	RF hard-wire control bus 8	**MD_GPIO2**	O:BPI_BUS8		
F34	**BPI_BUS9**	IO	RF hard-wire control bus 9	**MD_GPIO3**	O:BPI_BUS9		
			RF Serial Control Unit				
H32	**BSI_CS0**	IO	RF 3-wire interface chip select 0				
G33	**BSI_DATA**	IO	RF 3-wire interface data output				
J31	**BSI_CLK**	IO	RF 3-wire interface clock output				
			Analog Interface				
A3	**PAD_TDP1**	AIO	MIPI DSI data lane 1 +	GPIO4		I2S_DAT	
B4	**PAD_TDN1**	AIO	MIPI DSI data lane 1 -	GPIO5			
C1	**PAD_TDP0**	AIO	MIPI DSI data lane 0 +	GPIO2		I2S_WS	
D2	**PAD_TDN0**	AIO	MIPI DSI data lane 0 -	GPIO3		I2S_CLK	
B2	**PAD_TCP**	AIO	MIPI DSI clock lane +	GPIO6			
C3	**PAD_TCN**	AIO	MIPI DSI clock lane -	GPIO7			

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
H2	**PAD_RDP1**	AIO	MIPI DSI data lane 1 +	GPIO10			
G1	**PAD_RDN1**	AIO	MIPI DSI data lane 1 -	GPIO11			
E3	**PAD_RDP0**	AIO	MIPI DSI data lane 0 +	GPIO8		CLKM2	
E1	**PAD_RDN0**	AIO	MIPI DSI data lane 0 -	GPIO9		CLKM3	
G3	**PAD_RCP**	AIO	MIPI CSI clock lane +	GPIO12			
F2	**PAD_RCN**	AIO	MIPI CSI clock lane -	GPIO106			
E21	**PAD_FSRES**	AIO	PAD_FSRES				
G5	**PAD_TVRT**	AIO	PAD_TVRT				
A35	**PAD_AU_MOUTR**	AIO	Audio analog output right channel				
B34	**PAD_AU_MOUTL**	AIO	Audio analog output left channel				
C33	**PAD_AU_FMINR**	AIO	FM radio analog input right channel				
D34	**PAD_AU_FMINL**	AIO	FM radio analog input left channel				
A33	**PAD_AU_OUT0_P**	AIO	Earphone 0 amplifier output (+)				
B32	**PAD_AU_OUT0_N**	AIO	Earphone 0 amplifier output (-)				
H28	**PAD_AU_MICBIAS_P**	AIO	Microphone bias supply (+)				
A31	**PAD_AU_VCM_PO**	AIO	Audio output reference voltage (+)				
A29	**PAD_AU_VCM_NO**	AIO	Audio output reference voltage (-)				
D28	**PAD_AU_VIN0_P**	AIO	Microphone 0 amplifier input (+)				
C29	**PAD_AU_VIN0_N**	AIO	Microphone 0 amplifier input (-)				
D26	**PAD_AU_VIN1_P**	AIO	Microphone 1 amplifier input (+)				
C27	**PAD_AU_VIN1_N**	AIO	Microphone 1 amplifier input (-)				
E25	**PAD_BDLAQP**	AIO	Quadrature input (Q+) baseband codec downlink				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
C25	**PAD_BDLAQN**	AIO	Quadrature input (Q-) baseband codec downlink				
A27	**PAD_BDLAIP**	AIO	In-phase input (I+) baseband codec downlink				
B26	**PAD_BDLAIN**	AIO	In-phase input (I-) baseband codec downlink				
D24	**PAD_APC**	AIO	Automatic power control DAC output				
A21	**AUX_IN0_VBOUT**	AIO	Auxiliary ADC input 0				
J21	**PAD_AUX_IN1**	AIO	Auxiliary ADC input 1				
D20	**PAD_AUX_IN2**	AIO	Auxiliary ADC input 2				
G21	**AUX_IN3_ISENSE**	AIO	Auxiliary ADC input 3				
C21	**PAD_AUX_IN4**	AIO	Auxiliary ADC input 4				
H22	**AUX_IN5_CHRIN**	AIO	Auxiliary ADC input 5				
B22	**PAD_AUX_IN6**	AIO	Auxiliary ADC input 6				
F22	**PAD_AUX_IN7**	AIO	Auxiliary ADC input 7				
F18	**PAD_AUX_IN8**	AIO	Auxiliary ADC input 8				
E23	**PAD_AUX_XP**	AIO	Touch Panel X Plus(+) input				
D22	**PAD_AUX_YP**	AIO	Touch Panel Y Plus(+) input				
C23	**PAD_AUX_XM**	AIO	Touch Panel X Minus(-) input				
B24	**PAD_AUX_YM**	AIO	Touch Panel Y Minus(-) input				
F24	**PAD_AFC**	AIO	Automatic frequency control DAC output				
C19	**PAD_TVOUT**	AO	TV DAC current output				
D18	**PAD_SYSCLK**	AI	CLKSQ Input PAD				
J15	**PAD_USB_VBUS**	AIO	USB 5v power from USB host				
A15	**PAD_USB_ID**	AIO	USB ID pin for otg				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			device				
A13	**PAD_USB_DM**	AIO	USB D- Input/Output				
B14	**PAD_USB_DP**	AIO	USB D+ Input/Output				
H16	**PAD_USB_VRT**	AIO	for USB PHY bandgap reference				
			Image Sensor Interface				
AJ7	**CMRST**	IO	Image sensor reset signal output	**GPIO17**	O:CMRST	O:DSP2_GPO3	O:D1_TID0
AM4	**CMPDN**	IO	Image sensor power down control	**GPIO18**	O:CMPDN	O:DSP2_GPO2	O:D1_TID1
AH8	**CMVREF**	IO	Sensor vertical reference signal input	**GPIO19**	I:CMVREF	O:DSP_GPO3	O:TBTXEN
AT2	**CMHREF**	IO	Sensor horizontal reference signal input	**GPIO20**	I:CMHREF	O:DSP_GPO2	O:TBTXFS
AG9	**CMPCLK**	IO	Image sensor pixel clock input				
AR3	**CMMCLK**	IO	Image sensor master clock output				
AL7	**CMDAT9**	IO	Image sensor data input 9				
AP4	**CMDAT8**	IO	Image sensor data input 8				
AK8	**CMDAT7**	IO	Image sensor data input 7				
AN5	**CMDAT6**	IO	Image sensor data input 6				
AN7	**CMDAT5**	IO	Image sensor data input 5				
AU3	**CMDAT4**	IO	Image sensor data input 4				
AM8	**CMDAT3**	IO	Image sensor data input 3				
AT4	**CMDAT2**	IO	Image sensor data input 2				
AL9	**CMDAT1**	IO	Image sensor data input 1				
AV2	**CMDAT0**	IO	Image sensor data input 0				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AN9	**CMFLASH**	IO	Camera Flash Light Control	**GPIO33**	O:CMFLASH	O:D2_TID2	
R33	**SCL0**	IO	I2C clock		I2C Controller 2 (I2C2 SCL)		
AG5	**SCL1**	IO	I2C clock	**GPIO34**	I2C Controller 1 (I2C SCL)	O:D2_TID3	
AE31	**SCL2**	IO	I2C clock	**GPIO135**	I2C Controller 3 (I2C3 SCL)		
L37	**SDA0**	IO	I2C Data		I2C Controller 2 (I2C2 SDA)		
AP2	**SDA1**	IO	I2C Data	**GPIO35**	I2C Controller 1 (I2C SDA)	O:D2_TID4	
AL37	**SDA2**	IO	I2C Data	**GPIO136**	I2C Controller 3 (I2C3 SDA)		
			PWM Interface				
AH36	**PWM0**	IO	Pulse width modulated signal 0	**GPIO54**	O:PWM0	I:EINT16	
AM38	**PWM1**	IO	Pulse width modulated signal 1	**GPIO55**	O:PWM1	I:BSI_RFIN	I:EINT17
AR1	**PWM2**	IO	Pulse width modulated signal 2	**GPIO36**	O:PWM2	O:D2_TID5	
AG7	**PWM3**	IO	Pulse width modulated signal 3	**GPIO37**	O:PWM3	O:D2_TID6	CEVA_GPIO31
AK4	**PWM4**	IO	Pulse width modulated signal 4	**GPIO0**	O:PWM4	O:CLKM0	
AH6	**PWM5**	IO	Pulse width modulated signal 5	**GPIO99**	O:PWM5		
AJ37	**PWM6**	IO	Pulse width modulated signal 6	**GPIO24**	O:PWM6	O:DSP2_GPO0	
			Serial LCD/PM IC Interface				
U5	**LSCK**	IO	Serial display interface data output	**GPIO42**	O:LSCK	O:TDMA_CK	
U3	**LSA0**	IO	Serial display interface address output	**GPIO43**	O:LSA0	O:TDMA_D1	O:TDTIRQ
U1	**LSDA**	IO	Serial display interface clock output	**GPIO44**	LSDA	O:TDMA_D0	O:TCTIRQ2
V4	**LSCE0B**	IO	Serial display interface chip select 0 output	**GPIO45**	O:LSCE0B	O:TDMA_FS	O:TCTIRQ1

Ball	Name	Dir	Description				
38X38				**Mode0**	**Mode1**	**Mode2**	**Mode3**
V6	**LSCE1B**	IO	Serial display interface chip select 1 output	**GPIO46**	O:LSCE1B	O:LPCE2B	O:TEVTVAL

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			Parallel LCD/Nand-Flash Interface				
V2	**LPCE1B**	IO	Parallel display interface chip select 1 output	**GPIO47**	O:LPCE1B	O:D2_TID1	O:usb_probe_out[0]
W1	**LPCE0B**	IO	Parallel display interface chip select 0 output				
W3	**LPTE**	IO					
W7	**LRSTB**	IO	Parallel display interface Reset Signal				
Y2	**LRDB**	IO	Parallel display interface Read Strobe				
Y4	**LPA0**	IO	Parallel display interface address output				
AA1	**LWRB**	IO	Parallel display interface Write Strobe				
Y6	**NLD25**	IO	Parallel LCD/Nand-Flash Data 25				
AB4	**NLD24**	IO	Parallel LCD/Nand-Flash Data 24				
AE1	**NLD23**	IO	Parallel LCD/Nand-Flash Data 23				
AD2	**NLD22**	IO	Parallel LCD/Nand-Flash Data 22				
AA7	**NLD21**	IO	Parallel LCD/Nand-Flash Data 21				
AC3	**NLD20**	IO	Parallel LCD/Nand-Flash Data 20				
Y8	**NLD19**	IO	Parallel LCD/Nand-Flash Data 19				
AG1	**NLD18**	IO	Parallel LCD/Nand-Flash Data 18				

 © 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AA5	**NLD17**	IO	Parallel LCD/Nand-Flash Data 17				
AF2	**NLD16**	IO	Parallel LCD/Nand-Flash Data 16				
AB6	**NLD15**	IO	Parallel LCD/Nand-Flash Data 15				
AE3	**NLD14**	IO	Parallel LCD/Nand-Flash Data 14				
AB8	**NLD13**	IO	Parallel LCD/Nand-Flash Data 13				
AD4	**NLD12**	IO	Parallel LCD/Nand-Flash Data 12				
AC7	**NLD11**	IO	Parallel LCD/Nand-Flash Data 11				
AJ1	**NLD10**	IO	Parallel LCD/Nand-Flash Data 10				
AH2	**NLD9**	IO	Parallel LCD/Nand-Flash Data 9				
AL1	**NLD8**	IO	Parallel LCD/Nand-Flash Data 8				
AG3	**NLD7**	IO	Parallel LCD/Nand-Flash Data 7				
AF4	**NLD6**	IO	Parallel LCD/Nand-Flash Data 6				
AC5	**NLD5**	IO	Parallel LCD/Nand-Flash Data 5				
AK2	**NLD4**	IO	Parallel LCD/Nand-Flash Data 4				
AD6	**NLD3**	IO	Parallel LCD/Nand-Flash Data 3				
AJ3	**NLD2**	IO	Parallel LCD/Nand-Flash Data 2				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AD8	**NLD1**	IO	Parallel LCD/Nand-Flash Data 1				
AN1	**NLD0**	IO	Parallel LCD/Nand-Flash Data 0				
AE7	**NRNB**	IO	Nand-Flash Read/Busy Flag				
AH4	**NCLE**	IO	Nand-Flash Command Latch Signal				
AE5	**NALE**	IO	Nand-Flash Address Latch Signal				
AM2	**NWEB**	IO	Nand-Flash Write Strobe				
AF6	**NREB**	IO	Nand-Flash Read Strobe				
AJ5	**NCE0B**	IO	Nand-Flash Chip select output				
AL3	**NCE1B**	IO	Nand-Flash Chip select output				
			DPI Interface				
W5	**DPIHSYNC**	IO	DPI horizontal sync signal				
AA3	**DPIVSYNC**	IO	DPI vertical sync signal				
AB2	**DPIDE**	IO	DPI data enable signal				
AC1	**DPICK**	IO	DPI clock				
			Memory Card Interface				
L1	**MC0CM0**	IO	SD Command/MS Bus State Output				
T8	**MC0DA0**	IO	SD Serial Data IO 0/MS Serial Data IO				
N1	**MC0DA1**	IO	SD Serial Data IO 1				
R3	**MC0DA2**	IO	SD Serial Data IO 2				
P2	**MC0DA3**	IO	SD Serial Data IO 3				
U7	**MC0CK**	IO	SD Serial Clock/MS Serial Clock Output				
T4	**MC0PWRON**	IO	SD Power On Control Output				
R1	**MC0WP**	IO	SD Write Protect				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			Input				
T2	**MC0INS**	IO	SD Card Detect Input				
L3	**MC1CM0**	IO	SD Command/MS Bus State Output				
J3	**MC1DA0**	IO	SD Serial Data IO 0/MS Serial Data IO				
P6	**MC1DA1**	IO	SD Serial Data IO 1				
K2	**MC1DA2**	IO	SD Serial Data IO 0/MS Serial Data IO				
P4	**MC1DA3**	IO	SD Serial Data IO 1				
J1	**MC1CK**	IO	SD Serial Clock/MS Serial Clock Output				
R5	**MC1PWRON**	IO	SD Power On Control Output				
N3	**MC1WP**	IO	SD Write Protect Input				
M2	**MC1INS**	IO	SD Card Detect Input				
K6	**MC2CM0**	IO	SD Command/MS Bus State Output	**GPIO40**	MC2CM	MC0DA6	
L7	**MC2DA0**	IO	SD Serial Data IO 0/MS Serial Data IO				
L5	**MC2DA1**	IO	SD Serial Data IO 1				
K4	**MC2CK**	IO	SD Serial Clock/MS Serial Clock Output				
M6	**MC2PWRON**	IO	SD Power On Control Output	**GPIO39**	O:MC2PWRON	MC0DA5	
N5	**MC2WP**	IO	SD Write Protect Input	**GPIO38**	I:MC2WP	MC0DA4	
M4	**MC2INS**	IO	SD Card Detect Input	**GPIO41**	I:MC2INS	MC0DA7	
			Trace32 Interface				
AP30	**TRACECLK**	IO	Trace32 clock	**GPIO137**	O:TRACECLK		
AR31	**TRACECTL**	IO	Trace32 control signal	**GPIO138**	O:TRACECTL		
AU35	**TRACEDATA0**	IO	Trace32 data bus 0	**GPIO139**	O:TRACEDATA0		
AM28	**TRACEDATA1**	IO	Trace32 data bus 1	**GPIO140**	O:TRACEDATA1		
AT34	**TRACEDATA2**	IO	Trace32 data bus	**GPIO141**	O:TRACEDATA2		

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			2				
AN29	**TRACEDATA3**	IO	Trace32 data bus 3	**GPIO142**	O:TRACEDATA3		
AR33	**TRACEDATA4**	IO	Trace32 data bus 4	**GPIO143**	O:TRACEDATA4		
AL29	**TRACEDATA5**	IO	Trace32 data bus 5	**GPIO144**	O:TRACEDATA5		
AP32	**TRACEDATA6**	IO	Trace32 data bus 6	**GPIO145**	O:TRACEDATA6		
AM30	**TRACEDATA7**	IO	Trace32 data bus 7	**GPIO146**	O:TRACEDATA7		
			RTC Interface				
AB38	**XIN**	AO	32.768 KHz crystal input				
AC37	**XOUT**	AI	32.768 KHz crystal output				
AA35	**BBWAKEUP**	AIO	Baseband power on/off control				
AA33	**TESTMODE**	AI	TESTMODE enable input				
			External Memory Interface 1				
AP20	**EA26**	IO	External memory address bus 26				
AU19	**EA25**	IO	External memory address bus 25				
AV18	**EA24**	IO	External memory address bus 24				
AR19	**EA23**	IO	External memory address bus 23				
AM20	**EA22**	IO	External memory address bus 22				
AV22	**EA21**	IO	External memory address bus 21				
AR21	**EA20**	IO	External memory address bus 20				
AN19	**EA19**	IO	External memory address bus 19				
AV20	**EA18**	IO	External memory address bus 18				
AT20	**EA17**	IO	External memory address bus 17				
AU21	**EA16**	IO	External memory address bus 16				
AM16	**EA15**	IO	External memory address bus 15				
AU13	**EA14**	IO	External memory address bus 14				
AT22	**EA13**	IO	External memory				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			address bus 13				
AU15	**EA12**	IO	External memory address bus 12				
AT14	**EA11**	IO	External memory address bus 11				
AP14	**EA10**	IO	External memory address bus 10				
AR13	**EA9**	IO	External memory address bus 9				
AV14	**EA8**	IO	External memory address bus 8				
AV12	**EA7**	IO	External memory address bus 7				
AL13	**EA6**	IO	External memory address bus 6				
AN13	**EA5**	IO	External memory address bus 5				
AP12	**EA4**	IO	External memory address bus 4				
AM14	**EA3**	IO	External memory address bus 3				
AT12	**EA2**	IO	External memory address bus 2				
AU11	**EA1**	IO	External memory address bus 1				
AN15	**EA0**	IO	External memory address bus 0				
AU25	**EWAIT**	IO	External memory wait signal				
AT10	**EDQS3**	IO	External memory strobe signal 3				
AV6	**EDQS2**	IO	External memory strobe signal 2				
AV28	**EDQS1**	IO	External memory strobe signal 1				
AV32	**EDQS0**	IO	External memory strobe signal 0				
AV10	**EDQM3**	IO	External memory mask signal 3				
AU7	**EDQM2**	IO	External memory mask signal 2				
AU27	**EDQM1**	IO	External memory mask signal 1				
AU31	**EDQM0**	IO	External memory mask signal 0				
AU17	**ERAS_B**	IO	External memory row address select signal				
AN17	**ECAS_B**	IO	External memory				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			column address select signal				
AV26	**EADV_B**	IO	Flash, PSRAM and CellularRAM address valid				
AT18	**EC_CLK**	IO	Flash, PSRAM and CellularRAM clock				
AR17	**ED_CLK**	IO	MobileRAM clock				
AP18	**ED_CLK_B**	IO	MobileRAM clock				
AT16	**ECKE**	IO	MobileRAM clock enable				
AV16	**EWR_B**	IO	External memory write strobe				
AP22	**ERD_B**	IO	External memory read strobe				
AP16	**ECS0_B**	IO	External memory chip select 0				
AL17	**ECS1_B**	IO	External memory chip select 1				
AV24	**ECS2_B**	IO	External memory chip select 2				
AU23	**ECS3_B**	IO	External memory chip select 3				
AP10	**ED31**	IO	External memory data bus 31				
AN11	**ED30**	IO	External memory data bus 30				
AR9	**ED29**	IO	External memory data bus 29				
AM12	**ED28**	IO	External memory data bus 28				
AT8	**ED27**	IO	External memory data bus 27				
AU9	**ED26**	IO	External memory data bus 26				
AV8	**ED25**	IO	External memory data bus 25				
AR11	**ED24**	IO	External memory data bus 24				
AP6	**ED23**	IO	External memory data bus 23				
AT6	**ED22**	IO	External memory data bus 22				
AU5	**ED21**	IO	External memory data bus 21				
AR7	**ED20**	IO	External memory data bus 20				
AP8	**ED19**	IO	External memory				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
			data bus 19				
AM10	**ED18**	IO	External memory data bus 18				
AV4	**ED17**	IO	External memory data bus 17				
AL11	**ED16**	IO	External memory data bus 16				
AM22	**ED15**	IO	External memory data bus 15				
AN23	**ED14**	IO	External memory data bus 14				
AT26	**ED13**	IO	External memory data bus 13				
AR25	**ED12**	IO	External memory data bus 12				
AP24	**ED11**	IO	External memory data bus 11				
AV30	**ED10**	IO	External memory data bus 10				
AU29	**ED9**	IO	External memory data bus 9				
AT28	**ED8**	IO	External memory data bus 8				
AR27	**ED7**	IO	External memory data bus 7				
AT30	**ED6**	IO	External memory data bus 6				
AR29	**ED5**	IO	External memory data bus 5				
AV34	**ED4**	IO	External memory data bus 4				
AP28	**ED3**	IO	External memory data bus 3				
AU33	**ED2**	IO	External memory data bus 2				

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AT32	**ED1**	IO	External memory data bus 1				
AN27	**ED0**	IO	External memory data bus 0				
			CEVA Interface				
Y38	**CEVA_TCK**	IO	JTAG protocol TCK	**GPIO13**	I:CEVA_TCK		
W37	**CEVA_TMS**	IO	JTAG protocol TMS	**GPIO14**	I:CEVA_TMS		
V34	**CEVA_TDI**	IO	JTAG protocol TDI	**GPIO15**	I:CEVA_TDI		
W35	**CEVA_TDO**	IO	JTAG protocol TDO	**GPIO16**	CEVA_TDO		
V32	**CEVA_RTCK**	IO	JTAG protocol RTCK	**GPIO98**	CEVA_RTCK		
AM34	**BTDMP_DIN1**	IO	BTDMP transmit channel - serial data in	**GPIO92**	I:BTDMP_DIN1	CEVA_GPIO8	
AL33	**BTDMP_FSP1**	IO	BTDMP transmit channel - frame synchronization pulse	**GPIO93**	BTDMP_FSP1	CEVA_GPIO9	PHY_CLK
AT36	**BTDMP_CLK1**	IO	BTDMP transmit channel - clock	**GPIO94**	BTDMP_CLK1	CEVA_GPIO10	LINE_STATE0
AK32	**BTDMP_DOUT2**	IO	BTDMP receive channel - serial data out	**GPIO95**	O:BTDMP_DOU2	CEVA_GPIO11	LINE_STATE1
AN35	**BTDMP_FSP2**	IO	BTDMP receive channel - frame synchronization pulse	**GPIO96**	BTDMP_FSP2	CEVA_GPIO12	O:TRACEDATA14
AK34	**BTDMP_CLK2**	IO	BTDMP receive channel - clock	**GPIO97**	BTDMP_CLK2	CEVA_GPIO13	O:TRACEDATA15
AN3	**CAM_STROBE**	IO	Camera strobe	**GPIO29**	CAM_STROBE	D2ID	O:usb_probe_out[3]
AU1	**CAM_MECHSH1**	IO	Camera mechsh1	**GPIO31**	O:CAM_MECHSH1	O:D2_TID0	O:usb_probe_out[1]
AK6	**CAM_MECHSH0**	IO	Camera mechsh0	**GPIO30**	O:CAM_MECHSH0	I:D2IMS	O:usb_probe_out[2]
			Supply Voltages				
H8	**DVDD28_MIPITX**		Supply voltage of MIPI TX				
J9	**DVDD28_MIPITX**		Supply voltage of MIPI TX				
J5	**DVDD28_MIPIRX**		Supply voltage of MIPI RX				
J7	**DVDD12_MIPI**		Supply voltage of MIPI				
K8	**DVDD12_MIPI**		Supply voltage of MIPI				
H10	**DVSS28_MIPITX**		GND for MIPI TX				

 © 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
J11	**DVSS28_MIPITX**		GND for MIPI TX				
R9	**DVSS28_MIPIRX**		GND for MIPI RX				
M8	**DVSS12_MIPI**		GND for MIPI				
N9	**DVSS12_MIPI**		GND for MIPI				
AA15	VDDK		Supply voltage of internal logic				
AA23	VDDK		Supply voltage of internal logic				
AB14	VDDK		Supply voltage of internal logic				
AB24	VDDK		Supply voltage of internal logic				
AC13	VDDK		Supply voltage of internal logic				
AC25	VDDK		Supply voltage of internal logic				
AD12	VDDK		Supply voltage of internal logic				
AD14	VDDK		Supply voltage of internal logic				
AD16	VDDK		Supply voltage of internal logic				
AD18	VDDK		Supply voltage of internal logic				
AD20	VDDK		Supply voltage of internal logic				
AD22	VDDK		Supply voltage of internal logic				
AD24	VDDK		Supply voltage of internal logic				
AD26	VDDK		Supply voltage of internal logic				
AE15	VDDK		Supply voltage of internal logic				
AE21	VDDK		Supply voltage of internal logic				
AE23	VDDK		Supply voltage of internal logic				
AE27	VDDK		Supply voltage of internal logic				
AF14	VDDK		Supply voltage of internal logic				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AF18	VDDK		Supply voltage of internal logic				
AF24	VDDK		Supply voltage of internal logic				
AF28	VDDK		Supply voltage of internal logic				
AG13	VDDK		Supply voltage of internal logic				
AG17	VDDK		Supply voltage of internal logic				
AG21	VDDK		Supply voltage of internal logic				
AG25	VDDK		Supply voltage of internal logic				
AJ17	VDDK		Supply voltage of internal logic				
AJ21	VDDK		Supply voltage of internal logic				
AJ25	VDDK		Supply voltage of internal logic				
L13	VDDK		Supply voltage of internal logic				
M14	VDDK		Supply voltage of internal logic				
N15	VDDK		Supply voltage of internal logic				
P16	VDDK		Supply voltage of internal logic				
P22	VDDK		Supply voltage of internal logic				
P30	VDDK		Supply voltage of internal logic				
R15	VDDK		Supply voltage of internal logic				
R17	VDDK		Supply voltage of internal logic				
R21	VDDK		Supply voltage of internal logic				
R23	VDDK		Supply voltage of internal logic				
R25	VDDK		Supply voltage of internal logic				
R27	VDDK		Supply voltage of internal logic				
T14	VDDK		Supply voltage of internal logic				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
T18	VDDK		Supply voltage of internal logic				
T20	VDDK		Supply voltage of internal logic				
T24	VDDK		Supply voltage of internal logic				
U15	VDDK		Supply voltage of internal logic				
U23	VDDK		Supply voltage of internal logic				
V8	VDDK		Supply voltage of internal logic				
W13	VDDK		Supply voltage of internal logic				
W15	VDDK		Supply voltage of internal logic				
W23	VDDK		Supply voltage of internal logic				
Y12	VDDK		Supply voltage of internal logic				
AA17	VSS33		Ground of internal logic				
AA19	VSS33		Ground of internal logic				
AA21	VSS33		Ground of internal logic				
AA27	VSS33		Ground of internal logic				
AB12	VSS33		Ground of internal logic				
AB16	VSS33		Ground of internal logic				
AB18	VSS33		Ground of internal logic				
AB20	VSS33		Ground of internal logic				
AB22	VSS33		Ground of internal logic				
AC15	VSS33		Ground of internal logic				
AC17	VSS33		Ground of internal logic				
AC19	VSS33		Ground of internal logic				
AC21	VSS33		Ground of internal logic				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AC23	VSS33		Ground of internal logic				
AC27	VSS33		Ground of internal logic				
AE17	VSS33		Ground of internal logic				
AE19	VSS33		Ground of internal logic				
AF12	VSS33		Ground of internal logic				
AG15	VSS33		Ground of internal logic				
AG19	VSS33		Ground of internal logic				
AG23	VSS33		Ground of internal logic				
AG27	VSS33		Ground of internal logic				
AH14	VSS33		Ground of internal logic				
AH20	VSS33		Ground of internal logic				
AH24	VSS33		Ground of internal logic				
AK14	VSS33		Ground of internal logic				
AK20	VSS33		Ground of internal logic				
AK24	VSS33		Ground of internal logic				
AK26	VSS33		Ground of internal logic				
AL15	VSS33		Ground of internal logic				
AL19	VSS33		Ground of internal logic				
AL25	VSS33		Ground of internal logic				
M12	VSS33		Ground of internal logic				
M16	VSS33		Ground of internal logic				
M18	VSS33		Ground of internal logic				
M20	VSS33		Ground of internal logic				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
M22	VSS33		Ground of internal logic				
M24	VSS33		Ground of internal logic				
M26	VSS33		Ground of internal logic				
N27	VSS33		Ground of internal logic				
P12	VSS33		Ground of internal logic				
P18	VSS33		Ground of internal logic				
P20	VSS33		Ground of internal logic				
R19	VSS33		Ground of internal logic				
T16	VSS33		Ground of internal logic				
T22	VSS33		Ground of internal logic				
U17	VSS33		Ground of internal logic				
U19	VSS33		Ground of internal logic				
U21	VSS33		Ground of internal logic				
U27	VSS33		Ground of internal logic				
V14	VSS33		Ground of internal logic				
V16	VSS33		Ground of internal logic				
V18	VSS33		Ground of internal logic				
V20	VSS33		Ground of internal logic				
V22	VSS33		Ground of internal logic				
V24	VSS33		Ground of internal logic				
W17	VSS33		Ground of internal logic				
W19	VSS33		Ground of internal logic				
W21	VSS33		Ground of internal logic				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
W27	VSS33		Ground of internal logic				
Y14	VSS33		Ground of internal logic				
Y16	VSS33		Ground of internal logic				
Y18	VSS33		Ground of internal logic				
Y20	VSS33		Ground of internal logic				
Y22	VSS33		Ground of internal logic				
Y24	VSS33		Ground of internal logic				
AK28	VDD33_TRACE		Supply voltage of trace				
AL27	VDD33_TRACE		Supply voltage of trace				
AK10	VDD33_CAMERA		Supply voltage of camera				
AJ9	VDD33_CAMERA		Supply voltage of camera				
G11	VDD33_SPI		Supply voltage of SPI				
AA9	VDD33_NLD		Supply voltage of NLD				
AC9	VDD33_NLD		Supply voltage of NLD				
W9	VDD33_NLD		Supply voltage of NLD				
L11	VDD33_MIPI		Supply voltage of MIPI				
N7	VDD33_MC2		Supply voltage of memory card interface drivers				
U9	VDD33_MC1		Supply voltage of memory card interface drivers				
T6	VDD33_MC0		Supply voltage of memory card interface drivers				
AF8	VDD33_I2C		Supply voltage of I2C				
AK12	VDD33_EMI		Supply voltage of memory interface drivers				

© 2009 MediaTek Inc.

Ball 38X38	Name	Dir	Description	Mode0	Mode1	Mode2	Mode3
AK16	VDD33_EMI		Supply voltage of memory interface drivers				
AK18	VDD33_EMI		Supply voltage of memory interface drivers				
AK22	VDD33_EMI		Supply voltage of memory interface drivers				
AL21	VDD33_EMI		Supply voltage of memory interface drivers				
AL23	VDD33_EMI		Supply voltage of memory interface drivers				
AF30	VDD33		3.3v supply voltage				
AG31	VDD33		3.3v supply voltage				
AH30	VDD33		3.3v supply voltage				
K14	VDD33		3.3v supply voltage				
K28	VDD33		3.3v supply voltage				
M30	VDD33		3.3v supply voltage				
N31	VDD33		3.3v supply voltage				
T30	VDD33		3.3v supply voltage				
U31	VDD33		3.3v supply voltage				
AR23	FSOURCE						
			Analog Supply				
AB30	AVDD_RTC		Supply voltage for Real Time Clock				
B18	AVDD12_PLL		Supply voltage for PLL				
D16	AVDD12_USB		Supply voltage USB				
B30	AVDD28_AFE		Supply voltage for voice band receive section				
A37	AVDD28_MBUF		Supply Voltage for Audio band section				
G7	AVDD28_MIPITX		Supply voltage for MIPITX				
E19	AVDD28_PLL		Supply voltage for PLL				
J25	AVDD28_RFE		GND for baseband receive section, APC, AFC and AUXADC				
H20	AVDD28_TVDAC		TV DAC VDD				
AA37	AVDD30_VSIM2		Supply voltage for SIM2				

Ball	Name	Dir	Description				
38X38				**Mode0**	**Mode1**	**Mode2**	**Mode3**
W33	AVDD30_VSIM1		Supply voltage for SIM1				
F16	AVDD33_USB		Supply voltage USB				
A19	AVSS12_PLL		GND for PLL				
J17	AVSS12_USB		GND for USB				
B36	AVSS28_MBUF		GND for voice band transmit section				
B28	AVSS28_AFE		GND for voice band receive section				
L9	AVSS28_MIPITX		GND for MIPITX				
F20	AVSS28_PLL		GND for PLL				
H24	AVSS28_RFE		Supply voltage for baseband receive section, APC, AFC and AUXADC				
B20	AVSS28_TVDAC		TV DAC VSS				
W31	AVSS30_VSIM1		GND for VSIM1				
AB32	AVSS30_VSIM2		GND for VSIM2				
G15	AVSS33_USB		GND for USB				

© 2009 MediaTek Inc.

1.5 Power Description

Table 3 Power Descriptions

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	AVDD28_MBUF					
	AVDD28_MBUF					
B34	PAD_AU_MOUTL	AVDD28_AFE	AVSS28_AFE	AVDD28_AFE	AVSS28_AFE	
A35	PAD_AU_MOUTR					
	AVSS28_MBUF					
	AVSS28_MBUF					
C33	PAD_AU_FMINR	AVDD28_AFE	AVSS28_AFE	AVDD28_AFE	AVSS28_AFE	
D34	PAD_AU_FMINL					
A33	PAD_AU_OUT0_P					
B32	PAD_AU_OUT0_N					
H28	PAD_AU_MICBIAS_P					
	AVDD28_AFE					
	AVDD28_AFE					
	AVDD28_AFE					
	AVSS28_AFE					
A31	PAD_AU_VCM_PO	AVDD28_AFE	AVSS28_AFE	AVDD28_AFE	AVSS28_AFE	
A29	PAD_AU_VCM_NO					
D28	PAD_AU_VIN0_P					
C29	PAD_AU_VIN0_N					
D26	PAD_AU_VIN1_P					
C27	PAD_AU_VIN1_N					
	AVSS28_AFE					
	AVSS28_AFE					
A27	PAD_BDLAIP	AVDD28_RFE	AVSS28_RFE	AVDD28_RFE	AVSS28_RFE	
B26	PAD_BDLAIN					
E25	PAD_BDLAQP					
C25	PAD_BDLAQN					
	AVDD28_RFE					
	AVDD28_RFE					
	AVSS28_RFE					
	AVSS28_RFE					
	AVSS28_RFE					
D24	PAD_APC	AVDD28_RFE	AVSS28_RFE	AVDD28_RFE	AVSS28_RFE	
F24	PAD_AFC					
E23	PAD_AUX_XP					
D22	PAD_AUX_YP					
C23	PAD_AUX_XM					
B24	PAD_AUX_YM					
F18	PAD_AUX_IN8					
F22	PAD_AUX_IN7					
B22	PAD_AUX_IN6					

© 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
H22	AUX_IN5_CHRIN					
C21	PAD_AUX_IN4					
G21	AUX_IN3_ISENSE					
D20	PAD_AUX_IN2					
J21	PAD_AUX_IN1					
A21	AUX_IN0_VBOUT					
	AVSS28_TVDAC					
C19	PAD_TVOUT	AVDD28_TVDAC	AVSS28_TVDAC	AVDD28_TVDAC	AVSS28_TVDAC	
E21	PAD_FSRES					
	AVSS28_TVDAC					
	AVSS28_PLL					
D18	PAD_SYSCLK	AVDD28_PLL	AVSS28_PLL	AVDD28_PLL	AVSS28_PLL	
	AVDD28_PLL					
	AVDD12_PLL					
	AVSS12_PLL					
	AVDD12_PLL					
	AVSS12_PLL					
	AVDD12_PLL					
	AVSS12_PLL					
	AVDD12_PLL					
	VSS33					
	AVSS12_USB					
	AVDD12_USB					
	AVSS12_USB					
	AVDD12_USB					
A15	PAD_USB_ID	AVDD33_USB	AVSS33_USB	VDDK	VSS33	
	VSS33					
	AVDD33_USB					
H16	PAD_USB_VRT	AVDD33_USB	AVSS33_USB	VDDK	VSS33	
	AVDD33_USB					
	AVSS33_USB					
	AVDD33_USB					
	AVSS33_USB					
B14	PAD_USB_DP	AVDD33_USB	AVSS33_USB	VDDK	VSS33	
	AVSS33_USB					
A13	PAD_USB_DM	AVDD33_USB	AVSS33_USB	VDDK	VSS33	
J15	PAD_USB_VBUS					
	VSS33					
	VSS33					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
D14	USB_DRVVBUS	VDD33	VSS33	VDDK	VSS33	
C13	KCOL7					
	VSS33					

Ball 38X38	**Name**	**IO Supply**	**IO GND**	**Core Supply**	**Core GND**	**Remark**
B12	KCOL6	VDD33	VSS33	VDDK	VSS33	
A11	KCOL5					
	VSS33					
D12	KCOL4	VDD33	VSS33	VDDK	VSS33	
C11	KCOL3					
	VSS33					
B10	KCOL2	VDD33	VSS33	VDDK	VSS33	
A9	KCOL1					
	VSS33					
B8	KCOL0	VDD33	VSS33	VDDK	VSS33	
	VDD33					
	VSS33					
E11	KROW7	VDD33	VSS33	VDDK	VSS33	
	VSS33					
C9	KROW6	VDD33	VSS33	VDDK	VSS33	
D10	KROW5					
	VSS33					
E9	KROW4	VDD33	VSS33	VDDK	VSS33	
A7	KROW3					
	VSS33					
B6	KROW2	VDD33	VSS33	VDDK	VSS33	
D8	KROW1					
	VSS33					
F10	KROW0	VDD33	VSS33	VDDK	VSS33	
A5	SPI_CS_N					
	VSS33					
C5	SPI_SCK	VDD33	VSS33	VDDK	VSS33	
	VDD33_SPI					
	VSS33					
D6	SPI_MOSI	VDD33	VSS33	VDDK	VSS33	
	VSS33					
E7	SPI_MISO	VDD33	VSS33	VDDK	VSS33	
	VDD33_MIPI					
	VSS33					
	DVDD28_MIPITX					
	DVSS28_MIPITX					
A3	PAD_TDP1	VDD33_MIPI	VSS33	VDDK	VSS33	
B4	PAD_TDN1					
C1	PAD_TDP0					
D2	PAD_TDN0					
B2	PAD_TCP					
C3	PAD_TCN					
	DVDD28_MIPITX					
	DVSS28_MIPITX					
	DVDD28_MIPITX					
	DVSS28_MIPITX					

Ball 38X38	**Name**	**IO Supply**	**IO GND**	**Core Supply**	**Core GND**	**Remark**
	AVDD28_MIPITX					
	AVSS28_MIPITX					
G5	PAD_TVRT	VDD33_MIPI	VSS33	VDDK	VSS33	
	DVSS28_MIPITX					
	DVDD28_MIPITX					
	DVSS28_MIPITX					
	DVDD28_MIPITX					
H2	PAD_RDP1	VDD33_MIPI	VSS33	VDDK	VSS33	
G1	PAD_RDN1					
E3	PAD_RDP0					
E1	PAD_RDN0					
G3	PAD_RCP					
F2	PAD_RCN					
	DVSS28_MIPITX					
	DVDD28_MIPITX					
	VSS33					
	VDD33_MIPI					
	VSS33					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
K6	MC2CM0	VDD33_MC2	DVSS	VDDK	DVSS	
L7	MC2DA0					
	VSS33					
L5	MC2DA1	VDD33_MC2	DVSS	VDDK	DVSS	
	VDD33_MC2					
	VSS33					
K4	MC2CK	VDD33_MC2	DVSS	VDDK	DVSS	
	VSS33					
M6	MC2PWRON	VDD33_MC2	DVSS	VDDK	DVSS	
N5	MC2WP					
	VSS33					
M4	MC2INS	VDD33_MC2	DVSS	VDDK	DVSS	
L3	MC1CM0	VDD33_MC1	DVSS	VDDK	DVSS	
	VSS33_MC1					
J3	MC1DA0	VDD33_MC1	DVSS	VDDK	DVSS	
P6	MC1DA1					
	VSS33_MC1					
K2	MC1DA2	VDD33_MC1	DVSS	VDDK	DVSS	
	VDD33_MC1					
	VSS33					
P4	MC1DA3	VDD33_MC1	DVSS	VDDK	DVSS	
	VSS33					
J1	MC1CK	VDD33_MC1	DVSS	VDDK	DVSS	
R5	MC1PWRON					

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VSS33					
N3	MC1WP	VDD33_MC1	DVSS	VDDK	DVSS	
M2	MC1INS					
	VSS33					
L1	MC0CM0	VDD33_MC0	DVSS	VDDK	DVSS	
T8	MC0DA0					
	VSS33					
N1	MC0DA1	VDD33_MC0	DVSS	VDDK	DVSS	
R3	MC0DA2					
	VSS33					
	VDD33_MC0					
	VSS33					
	VSS33					
P2	MC0DA3	VDD33_MC0	DVSS	VDDK	DVSS	
U7	MC0CK					
	VSS33					
T4	MC0PWRON	VDD33_MC0	DVSS	VDDK	DVSS	
R1	MC0WP					
	VSS33					
T2	MC0INS	VDD33_MC0	DVSS	VDDK	DVSS	
	VDDK					
	VSS33					
	VDDK					
	VSS33					
U5	LSCK	VDD33	VSS33	VDDK	VSS33	
	VSS33					
U3	LSA0	VDD33	VSS33	VDDK	VSS33	
U1	LSDA					
	VSS33					
V4	LSCE0B	VDD33	VSS33	VDDK	VSS33	
V6	LSCE1B					
	VSS33					
	VDD33_NLD					
	VSS33					
	VSS33					
W1	LPCE0B	VDD33	VSS33	VDDK	VSS33	
V2	LPCE1B					
	VSS33					
W3	LPTE	VDD33	VSS33	VDDK	VSS33	
W7	LRSTB					
	VSS33					
Y2	LRDB	VDD33	VSS33	VDDK	VSS33	
Y4	LPA0					
	VSS33					
AA1	LWRB	VDD33	VSS33	VDDK	VSS33	
W5	DPIHSYNC					

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VSS33					
AA3	DPIVSYNC	VDD33	VSS33	VDDK	VSS33	
	VDD33_NLD					
	VSS33					
AB2	DPIDE	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AC1	DPICK	VDD33	VSS33	VDDK	VSS33	
Y6	NLD25	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AB4	NLD24	VDD33_NLD	VSS33	VDDK	VSS33	
AE1	NLD23	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AD2	NLD22	VDD33_NLD	VSS33	VDDK	VSS33	
AA7	NLD21	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AC3	NLD20	VDD33_NLD	VSS33	VDDK	VSS33	
Y8	NLD19	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
	VDD33_NLD					
	VSS33					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
AG1	NLD18	VDD33_NLD	VSS33	VDDK	VSS33	
AA5	NLD17	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AF2	NLD16	VDD33_NLD	VSS33	VDDK	VSS33	
AB6	NLD15	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AE3	NLD14	VDD33_NLD	VSS33	VDDK	VSS33	
AB8	NLD13	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AD4	NLD12	VDD33_NLD	VSS33	VDDK	VSS33	
AC7	NLD11	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AJ1	NLD10	VDD33_NLD	VSS33	VDDK	VSS33	
AH2	NLD9	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
	VDD33_NLD					
	VSS33					
AL1	NLD8	VDD33_NLD	VSS33	VDDK	VSS33	
AG3	NLD7	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AF4	NLD6	VDD33_NLD	VSS33	VDDK	VSS33	
AC5	NLD5	VDD33_NLD	VSS33	VDDK	VSS33	

 © 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VSS33					
AK2	NLD4	VDD33_NLD	VSS33	VDDK	VSS33	
AD6	NLD3	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AJ3	NLD2	VDD33_NLD	VSS33	VDDK	VSS33	
AD8	NLD1	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AN1	NLD0	VDD33_NLD	VSS33	VDDK	VSS33	
AE7	NRNB	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
	VDD33_NLD					
	VSS33					
AH4	NCLE	VDD33_NLD	VSS33	VDDK	VSS33	
AE5	NALE	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AM2	NWEB	VDD33_NLD	VSS33	VDDK	VSS33	
AF6	NREB	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
AL3	NCE1B	VDD33_NLD	VSS33	VDDK	VSS33	
AJ5	NCE0B	VDD33_NLD	VSS33	VDDK	VSS33	
	VSS33					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
AP2	SDA1	VDD33_I2C	VSS33	VDDK	VSS33	
	VDD33_I2C					
	VSS33					
AG5	SCL1	VDD33_I2C	VSS33	VDDK	VSS33	
	VSS33					
AR1	PWM2	VDD33_CAMERA	VSS33	VDDK	VSS33	
AG7	PWM3	VDD33_CAMERA	VSS33	VDDK	VSS33	
	VSS33					
AK4	PWM4	VDD33_CAMERA	VSS33	VDDK	VSS33	
AH6	PWM5	VDD33_CAMERA	VSS33	VDDK	VSS33	
	VSS33					
	VDD33_CAMERA					
	VSS33					
AN3	CAM_STROBE	VDD33_CAMERA	VSS33	VDDK	VSS33	
AK6	CAM_MECHSH0					
	VSS33					
AU1	CAM_MECHSH1	VDD33_CAMER	VSS33	VDDK	VSS33	

© 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
AJ7	CMRST	A				
	VSS33					
AM4	CMPDN	VDD33_CAMER	VSS33	VDDK	VSS33	
AH8	CMVREF	A				
	VSS33					
AT2	CMHREF	VDD33_CAMER	VSS33	VDDK	VSS33	
AG9	CMPCLK	A				
	VSS33					
	VDD33_CAMERA					
	VDD33_CAMERA					
	VSS33					
AR3	CMMCLK	VDD33_CAMER	VSS33	VDDK	VSS33	
AL7	CMDAT9	A				
	VSS33					
AP4	CMDAT8	VDD33_CAMER	VSS33	VDDK	VSS33	
AK8	CMDAT7	A				
	VSS33					
AN5	CMDAT6	VDD33_CAMER	VSS33	VDDK	VSS33	
AN7	CMDAT5	A				
	VSS33					
AU3	CMDAT4	VDD33_CAMER	VSS33	VDDK	VSS33	
AM8	CMDAT3	A				
	VSS33					
	VDD33_CAMERA					
	VSS33					
AT4	CMDAT2	VDD33_CAMER	VSS33	VDDK	VSS33	
AL9	CMDAT1	A				
	VSS33					
AV2	CMDAT0	VDD33_CAMER	VSS33	VDDK	VSS33	
AN9	CMFLASH	A				
	VSS33					
AP6	ED23	VDD33_EMI	VSS33	VDDK	VSS33	
AT6	ED22					
	VSS33					
AU5	ED21	VDD33_EMI	VSS33	VDDK	VSS33	
	VDD33_EMI					
	VSS33					
AR7	ED20	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
AP8	ED19	VDD33_EMI	VSS33	VDDK	VSS33	
AM10	ED18					
	VSS33					
AV4	ED17	VDD33_EMI	VSS33	VDDK	VSS33	
AL11	ED16					
	VSS33					
AU7	EDQM2	VDD33_EMI	VSS33	VDDK	VSS33	

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VDDK					
	VSS33					
	VDDK					
	VSS33					
	VSS33					
AV6	EDQS2	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
AP10	ED31	VDD33_EMI	VSS33	VDDK	VSS33	
AN11	ED30					
	VSS33					
AR9	ED29	VDD33_EMI	VSS33	VDDK	VSS33	
AM12	ED28					
	VSS33					
AT8	ED27	VDD33_EMI	VSS33	VDDK	VSS33	
	VDD33_EMI					
	VSS33					
AU9	ED26	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
AV8	ED25	VDD33_EMI	VSS33	VDDK	VSS33	
AR11	ED24					
	VSS33					
AV10	EDQM3	VDD33_EMI	VSS33	VDDK	VSS33	
AT10	EDQS3					
	VSS33					
AP12	EA4	VDD33_EMI	VSS33	VDDK	VSS33	
	VDD33_EMI					
	VSS33					
AL13	EA6	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
AT12	EA2	VDD33_EMI	VSS33	VDDK	VSS33	
AN13	EA5					
	VSS33					
AR13	EA9	VDD33_EMI	VSS33	VDDK	VSS33	
AM14	EA3					
	VSS33					
AU11	EA1	VDD33_EMI	VSS33	VDDK	VSS33	
	VDD33_EMI					
	VSS33					
AV12	EA7	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
AU13	EA14	VDD33_EMI	VSS33	VDDK	VSS33	
AP14	EA10					
	VSS33					
AT14	EA11	VDD33_EMI	VSS33	VDDK	VSS33	
AN15	EA0					
	VSS33					

© 2009 MediaTek Inc.

Ball 38X38	**Name**	**IO Supply**	**IO GND**	**Core Supply**	**Core GND**	**Remark**
AV14	EA8	VDD33_EMI	VSS33	VDDK	VSS33	
	VSS33					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
AU15	EA12	VDD33_EMI	VSS33	VDDK	VSS33	
AM16	EA15					
	VSS33					
AP16	ECS0_B	VDD33_EMI	VSS33	VDDK	VSS33	
AL17	ECS1_B					
	VSS33					
AT16	ECKE	VDD33_EMI	VSS33	VDDK	VSS33	
AV16	EWR_B					
	VSS33					
AU17	ERAS_B	VDD33_EMI	VSS33	VDDK	VSS33	
AN17	ECAS_B					
	VSS33					
	VDD33_EMI					
	VSS33					
AR17	ED_CLK	VDD33_EMI	VSS33	VDDK	VSS33	
AP18	ED_CLK_B					
	VSS33					
	VDD33_EMI					
	VSS33					
AT18	EC_CLK	VDD33_EMI	VSS33	VDDK	VSS33	
AR19	EA23					
	VSS33					
AV18	EA24	VDD33_EMI	VSS33	VDDK	VSS33	
AN19	EA19					
	VSS33					
AU19	EA25	VDD33_EMI	VSS33	VDDK	VSS33	
AM20	EA22					
	VSS33					
AT20	EA17	VDD33_EMI	VSS33	VDDK	VSS33	
AV20	EA18					
	VSS33					
	VDD33_EMI					
	VSS33					
AU21	EA16	VDD33_EMI	VSS33	VDDK	VSS33	
AP20	EA26					
	VSS33					
AV22	EA21	VDD33_EMI	VSS33	VDDK	VSS33	
AR21	EA20					
	VSS33					
AT22	EA13	VDD33_EMI	VSS33	VDDK	VSS33	

 © 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
AU23	ECS3_B					
AR23	FSOURCE					
	VSS33					
	VDD33_EMI					
	VSS33					
	VDDK33					
	VSS33					
	VDDK33					
	VSS33					
AV24	ECS2_B	VDD33_EMI	VSS33	VDDK	VSS33	
AN21	WATCHDOG					
	VSS33					
AV26	EADV_B	VDD33_EMI	VSS33	VDDK	VSS33	
AP22	ERD_B					
	VSS33					
AU25	EWAIT	VDD33_EMI	VSS33	VDDK	VSS33	
AT24	EADMUX					
	VSS33					
	VDD33_EMI					
	VSS33					
AV28	EDQS1	VDD33_EMI	VSS33	VDDK	VSS33	
AU27	EDQM1					
	VSS33					
AM22	ED15	VDD33_EMI	VSS33	VDDK	VSS33	
AN23	ED14					
	VSS33					
AT26	ED13	VDD33_EMI	VSS33	VDDK	VSS33	
AR25	ED12					
	VSS33					
	VDD33_EMI					
	VSS33					
AP24	ED11	VDD33_EMI	VSS33	VDDK	VSS33	
AV30	ED10					
	VSS33					
AU29	ED9	VDD33_EMI	VSS33	VDDK	VSS33	
AT28	ED8					
	VSS33					
AV32	EDQS0	VDD33_EMI	VSS33	VDDK	VSS33	
AU31	EDQM0					
	VSS33					
	VDD33_EMI					
	VSS33					
AR27	ED7	VDD33_EMI	VSS33	VDDK	VSS33	
AT30	ED6					
	VSS33					
AR29	ED5	VDD33_EMI	VSS33	VDDK	VSS33	

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
AV34	ED4					
	VSS33					
AP28	ED3	VDD33_EMI	VSS33	VDDK	VSS33	
AU33	ED2					
	VSS33					
	VDD33_EMI					
	VSS33					
	VDDK33					
	VSS33					
	VDDK33					
	VSS33					
AT32	ED1	VDD33_EMI	VSS33	VDDK	VSS33	
AN27	ED0					
	VSS33					
AV36	MFIQ	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AP30	TRACECLK	VDD33_TRACE	VSS33	VDDK	VSS33	
AR31	TRACECTL					
	VSS33					
AU35	TRACEDATA0	VDD33_TRACE	VSS33	VDDK	VSS33	
AM28	TRACEDATA1					
	VSS33					
	VDD33_TRACE					
	VDD33_TRACE					
	VSS33					
AT34	TRACEDATA2	VDD33_TRACE	VSS33	VDDK	VSS33	
AN29	TRACEDATA3					
	VSS33					
AR33	TRACEDATA4	VDD33_TRACE	VSS33	VDDK	VSS33	
	VDD33_TRACE					
	VSS33					
	VDD33_TRACE					
AL29	TRACEDATA5	VDD33_TRACE	VSS33	VDDK	VSS33	
	VSS33					
AP32	TRACEDATA6	VDD33_TRACE	VSS33	VDDK	VSS33	
AM30	TRACEDATA7					
	VSS33					
AM32	J2TRST_B	VDD33	VSS33	VDDK	VSS33	
AK30	J2TCK					
	VSS33					
AN33	J2TDI	VDD33	VSS33	VDDK	VSS33	
AP34	J2TMS					
	VSS33					
AR35	J2TDO	VDD33	VSS33	VDDK	VSS33	
AL31	J2RTCK					
	VSS33					

© 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
AM34	BTDMP_DIN1	VDD33	VSS33	VDDK	VSS33	
	VDD33					
	VSS33					
AL33	BTDMP_FSP1	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AT36	BTDMP_CLK1	VDD33	VSS33	VDDK	VSS33	
Ak32	BTDMP_DOUT2					
	VSS33					
AN35	BTDMP_FSP2	VDD33	VSS33	VDDK	VSS33	
AK34	BTDMP_CLK2					
	VSS33					
AU37	GPIO123	VDD33	VSS33	VDDK	VSS33	
AJ33	DAIPCMOUT					
	VSS33					
AP36	DAIPCMIN	VDD33	VSS33	VDDK	VSS33	
AH32	DAISYNC					
	VSS33					
AR37	DAICLK	VDD33	VSS33	VDDK	VSS33	
AH34	UTXD2					
	VSS33					
AL35	URXD2	VDD33	VSS33	VDDK	VSS33	
	VDDK					
	VSS33					
	VDDK					
	VSS33					
	VDD33					
	VSS33					
AG33	URTS2	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AM36	UCTS2	VDD33	VSS33	VDDK	VSS33	
AF34	GPIO122	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AT38	DAIRST	VDD33	VSS33	VDDK	VSS33	
AN37	EINT1					
	VSS33					
AJ35	SRCLKENAI	VDD33	VSS33	VDDK	VSS33	
AF32	EINT2					
	VSS33					
AK36	EINT3	VDD33	VSS33	VDDK	VSS33	
AG35	EINT4					
	VSS33					
AP38	EINT5	VDD33	VSS33	VDDK	VSS33	
AE31	SCL2					
	VSS33					
AL37	SDA2	VDD33	VSS33	VDDK	VSS33	
AH36	PWM0					

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VSS33					
	VDD33					
	VSS33					
AM38	PWM1	VDD33	VSS33	VDDK	VSS33	
AJ37	PWM6					
	VSS33					
AK38	URXD3	VDD33	VSS33	VDDK	VSS33	
AD30	UTXD3					
	VSS33					
AE35	URXD4	VDD33	VSS33	VDDK	VSS33	
AE33	UTXD4					
	VSS33					
AF36	GPIO116	VDD33	VSS33	VDDK	VSS33	
AG37	GPIO117					
	VSS33					
AH38	GPIO118	VDD33	VSS33	VDDK	VSS33	
AD32	GPIO119					
	VSS33					
AD36	GPIO120	VDD33	VSS33	VDDK	VSS33	
AE37	GPIO121					
	VSS33					
AF38	GPIO124	VDD33	VSS33	VDDK	VSS33	
	VDDK					
	VSS33					
	VDDK					
	VSS33					
	VDD33					
	VSS33					
AC31	GPIO125	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AD34	JTRST_B	VDD33	VSS33	VDDK	VSS33	
AD38	JTCK					
	VSS33					
AB36	JTDI	VDD33	VSS33	VDDK	VSS33	
AC33	JTMS					
	VSS33					
AC35	JTDO	VDD33	VSS33	VDDK	VSS33	
AB34	JRTCK					
	VSS33					
	VSS33					
AC37	XOUT	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AB38	XIN	VDD33	VSS33	VDDK	VSS33	
	AVDD_RTCK					
AA33	TESTMODE	VDD33	VSS33	VDDK	VSS33	
	AVDD_RTCK					

© 2009 MediaTek Inc.

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
AA35	BBWAKEUP	VDD33	VSS33	VDDK	VSS33	
	VSS33					
AA31	PAD_SIO2	VDD33	VSS33	VDDK	VSS33	
Y34	PAD_SRST2					
Y30	PAD_SCLK2					
	AVDD30_VSIM2					
	AVDD30_VSIM2					
	AVDD30_VSIM					
	AVDD30_VSIM					
V30	PAD_SIO	VDD33	VSS33	VDDK	VSS33	
Y36	PAD_SRST					
Y32	PAD_SCLK					
	VDDK					
	VSS33					
	VDDK					
	VSS33					
Y38	CEVA_TCK	VDD33	VSS33	VDDK	VSS33	
	VSS33					
W37	CEVA_TMS	VDD33	VSS33	VDDK	VSS33	
V34	CEVA_TDI					
	VSS33					
W35	CEVA_TDO	VDD33	VSS33	VDDK	VSS33	
V32	CEVA_RTCK					
	VSS33					
V38	IBOOT	VDD33	VSS33	VDDK	VSS33	
V36	NC					
	VSS33					
	VDD33					
	VSS33					
U37	SECU_EN	VDD33	VSS33	VDDK	VSS33	
U33	ICORESIGHT					
	VSS33					
T38	IONEJTAG	VDD33	VSS33	VDDK	VSS33	
U35	HDQ					
	VSS33					
P38	SRCLKENAN	VDD33	VSS33	VDDK	VSS33	
	VSS33					
T36	GPIO126	VDD33	VSS33	VDDK	VSS33	
T32	GPIO127					
	VSS33					
R37	GPIO128	VDD33	VSS33	VDDK	VSS33	
M38	GPIO129					
	VSS33					
N37	GPIO130	VDD33	VSS33	VDDK	VSS33	
P34	GPIO131					
	VSS33					

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
P36	GPIO132	VDD33	VSS33	VDDK	VSS33	
	VDD33					
	VSS33					
R35	GPIO133	VDD33	VSS33	VDDK	VSS33	
	VSS33					
K38	PWR_KEY	VDD33	VSS33	VDDK	VSS33	
R33	SCL0					
	VSS33					
L37	SDA0	VDD33	VSS33	VDDK	VSS33	
M36	EINT0					
	VSS33					
H38	GPIO115	VDD33	VSS33	VDDK	VSS33	
N33	SRCLKENA					
	VSS33					
J37	SYSRST_B	VDD33	VSS33	VDDK	VSS33	
N35	EINT6					
	VSS33					
K36	EINT7	VDD33	VSS33	VDDK	VSS33	
P32	EINT8					
	VSS33					
L35	EINT9	VDD33	VSS33	VDDK	VSS33	
F38	I2S_WS					
	VSS33					
G37	I2S_CLK	VDD33	VSS33	VDDK	VSS33	
	VDDK					
	VSS33					
	VDDK					
	VSS33					
	VDD33					
	VSS33					
H36	I2S_DAT	VDD33	VSS33	VDDK	VSS33	
	VSS33					
D38	URXD1	VDD33	VSS33	VDDK	VSS33	
M34	UTXD1					
	VSS33					
E37	UCTS1	VDD33	VSS33	VDDK	VSS33	
J35	URTS1					
	VSS33					
F36	IRDA_RXD	VDD33	VSS33	VDDK	VSS33	
M32	IRDA_TXD					
	VSS33					
B38	IRDA_PDN	VDD33	VSS33	VDDK	VSS33	
L33	BPI_BUS0					
	VSS33					
H34	BPI_BUS1	VDD33	VSS33	VDDK	VSS33	
G35	BPI_BUS2					

Ball 38X38	Name	IO Supply	IO GND	Core Supply	Core GND	Remark
	VSS33					
C37	BPI_BUS3	VDD33	VSS33	VDDK	VSS33	
K34	BPI_BUS4					
	VSS33					
D36	BPI_BUS5	VDD33	VSS33	VDDK	VSS33	
	VDD33					
	VSS33					
K32	BPI_BUS6	VDD33	VSS33	VDDK	VSS33	
	VSS33					
E35	BPI_BUS7	VDD33	VSS33	VDDK	VSS33	
J33	BPI_BUS8					
	VSS33					
F34	BPI_BUS9	VDD33	VSS33	VDDK	VSS33	
H32	BSI_CS0					
	VSS33					
G33	BSI_DATA	VDD33	VSS33	VDDK	VSS33	
J31	BSI_CLK					

1.6 Ordering information

1.6.1 MT6516

Part number	Package	Operational temperature range
MT6516	15x15x1.2 mm 564-TFBGA	-20~80°C

Table 4 MT6516 Ordering Information

2 Application Micro-Controller Unit Subsystem

Figure 2-1 illustrates the block diagram of the Micro-Controller Unit Subsystem in MT6516. The subsystem utilizes a main 32-bit ARM926EJ-S RISC processor, which plays the role of the main bus master controlling the whole subsystem. The ARM926EJ-S RISC is equipped with instruction cache, instruction TCM, data cache, and data TCM. Both instruction and data cache have 32KB and the size of all TCM is 16KB. If the requested content is found in TCM or in cache, no bus transaction is required. If the code cache hit rate is high enough, bus traffic can be effectively reduced and processor core performance maximized.

The bus comprises of two-level system buses: Advanced High-Performance Bus (AHB) and Advanced Peripheral Bus (APB). All bus transactions originate from bus masters, while slaves can only respond to requests from bus masters. Before data transfer can be established, the bus master must ask for bus ownership, accomplished by request-grant handshaking protocol between masters and arbiters.

Two levels of bus hierarchy are designed to provide optimum usage for different performance requirements. Specifically, AHB Bus, the main system bus, is tailored toward high-speed requirements and provides 32-bit data path with multiplex scheme for bus interconnections. The APB Bus, on the other hand, is designed to reduce interface complexity for lower data transfer rate, and so it is isolated from high bandwidth AHB Bus by APB Bridge. APB Bus supports 16-bit addressing and both 16-bit and 32-bit data paths. APB Bus is also optimized for minimal power consumption by turning off the clock when there is no APB bus activity.

During operation, if the target slave is located on AHB Bus, the transaction is conducted directly on AHB Bus. However, if the target slave is a peripheral and is attached to the APB bus, then the transaction is conducted between AHB and APB bus through the use of APB Bridge.

The MT6516 MCU subsystem supports only memory addressing method. Therefore all components are mapped onto the MCU 32-bit address space.

In order to off-load the processor core, a DMA Controller is designated to act as a master and share the bus resources on AHB Bus to perform fast data movement between modules. This controller provides eleven DMA channels.

The Interrupt Controller provides a software interface to manipulate interrupt events; it can handle up to 64 interrupt sources asserted at the same time. In general, the controller generates 2 levels of interrupt requests, FIQ and IRQ, to the processor.

A 304K Byte SRAM is provided as system memory for high-speed data access. For factory programming purposes, a Boot ROM module is also integrated.

External Memory Interface supports 8-bit, 16-bit and 32-bit devices. This interface supports both synchronous and asynchronous components, such as Flash, SRAM and SDR, DDR SDRAM. This interface also supports page and burst mode type of Flash, Cellular RAM, as well as high performance MobileRAM. Since AHB Bus is 32-bit wide, all data transfers are converted into several 8-bit or 16-bit cycles depending on the data width of the target device. In contrast to code cache, contents in data cache are queried when MCU issues data requests, or when other AHB bus masters issue memory requests to EMI.

 © 2009 MediaTek Inc.

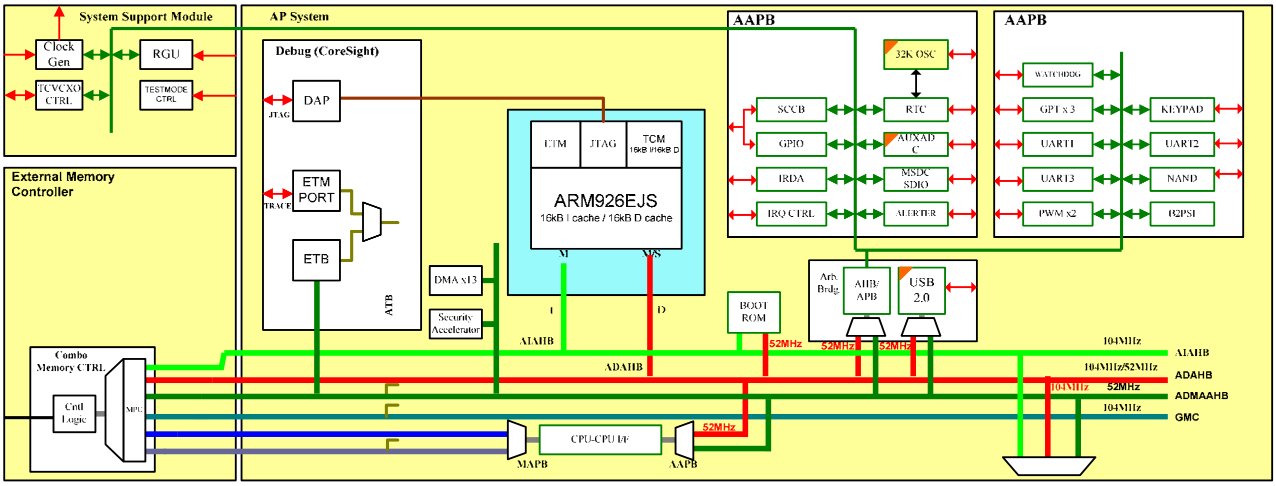


Figure 2-1 ***Block Diagram of the Micro-Controller Unit Subsystem in MT6516***

2.1 Processor Core

2.1.1 General Description

The Micro-Controller Unit Subsystem in MT6516 uses the 32-bit ARM926EJ-S RISC processor that is based on the Harvard architecture with two separated 32-bit data buses that carry instructions and data independently. The running clock frequency is up to 416MHz, 4 times the speed of the AHB bus. The memory interface of ARM926EJ-S is totally compliant with the AMBA based bus system, which allows direct connection to the AHB Bus.

2.1.2 General Programming Guide

2.1.2.1 Idle insertion between operations

In MT6516, the CPU runs at 416 MHz in default, which is 4 times faster than the connected 104 MHz AHB buses. Therefore, only one clock cycle of the outside system bus passes while CPU executes 4 instructions (assuming no stall, branch or abort). For example, if you insert 4 NOPs between two single-word memory write, you have chances to see the two AHB writes are consecutive on AHB bus. This must be noticed since certain codes would intentionally insert idle cycles between two operations and the absolute time of idleness may vary with the CPU clock speed. Let's assume that you used to insert 8 NOPs to separate two AHB operations in earlier product, which has CPU running at 208 MHz. The truth is the two operations would appear at least four bus clock cycles away from each other. But in MT6516 it may have only two idle cycles injected on the bus.

2.1.3 ARM926EJ-S Power Down

To stop the clock and make the ARM926EJ-S sleep, the following step should be taken:

1. Make sure the CPU is in the privileged mode.
2. Execute the wait for interrupt instruction.

 © 2009 MediaTek Inc.

MCR p15, 0, <Rd>, c7, c0, 4

The ARM926EJ-S core will be put into the sleep mode now and will be waken up by external FIQ or IRQ.

2.2 Memory Management

2.2.1 General Description

The processor core of MT6516 supports only a memory addressing method for instruction fetch and data access. The core manages a 32-bit address space that has addressing capability of up to 4 GB. System RAM, System ROM, Registers, MCU Peripherals and external components are all mapped onto such 32-bit address space, as depicted in **Figure 2-2**.

AP MCUSYS		
BANK	Base Address	Description
BANK0	0000_0000h	EMI Band 0 / Boot Code
BANK1	1000_0000h	EMI Bank 1
BANK2	2000_0000h	EMI Bank 2
BANK3	3000_0000h	EMI Bank 3
BANK4	4000_0000h	System RAM
	4800_0000h	System ROM
BANK5	5000_0000h	TCM
BANK6		Not Used
BANK7		Not Used
BANK8	8000_0000h	APB Peripheral
	8010_0000h	USB
	8011_0000h	Virtual FIFO Slave
	8012_0000h	LCD
	8013_0000h	DPI
	8014_0000h	DSI
BANK9		Not Used
BANK10		Not Used
BANK11	B000_0000h	CEVA
BANK12		Not Used
BANK13		Not Used
BANK14		Not Used
BANK15	F000_0000h	Back Door

Figure 2-2 ***Memory Layout of MT6516***

The address space is organized into blocks of 256 MB each. The block number is uniquely selected by address line A31-A28 of the internal system bus.

 © 2009 MediaTek Inc.

2.2.1.1 External Access

To allow external access, the MT6516 outputs 27 bits (A26-A0) of address lines along with 4 selection signals that correspond to associated memory blocks. That is, MT6516 can support up to 4 MCU addressable external components. The data width of internal system bus is fixed at 32-bit wide, while the data width of the external components can be 8-, 16- or 32- bit.

Since devices are usually available with varied operating grades, adaptive configurations for different applications are needed. MT6516 provides software programmable registers to configure their wait-states to adapt to different operating conditions.

2.2.1.2 Memory Re-mapping Mechanism

To permit more flexible system configuration, a memory re-mapping mechanism is provided. The mechanism allows software program to swap BANK0 (ECS0#) and BANK1 (ECS1#) dynamically. Whenever the bit value of RM0 in register EMI_REMAP is changed, these two banks are swapped accordingly. Furthermore, it allows system to boot from System ROM as detailed in 2.2.1.3 Boot Sequence.

2.2.1.3 Boot Sequence

Since the ARM926EJ-S core always starts to fetch instructions from the lowest memory address at 00000000h after system has been reset, the system is designed to have a dynamic mapping architecture capable of associating Boot Code, external Flash or external SRAM with the memory block 0000_0000h – 0fff_ffffh.

By default, the Boot Code is mapped onto 0000_0000h – 0fff_ffffh after a system reset. In this special boot mode, External Memory Controller does not access external memory; instead, the EMI Controller send predefined Boot Code back to the ARM926EJ-S core, which instructs the processor to execute the program in System ROM. This configuration can be changed by programming bit value of RM1 in register EMI_REMAP directly.

MT6516 system provides one boot up scheme:

- Start up system of running codes from Boot Code for factory programming or NAND flash boot.

2.2.1.3.1 Boot Code

The Boot Code is placed together with Memory Re-Mapping Mechanism in External Memory Controller, and comprises of just two words of instructions as shown below. A jump instruction leads the processor to run the code starting at address 4800_0000h where the System ROM is placed.

ADDRESS	BINARY CODE	ASSEMBLY	
00000000h	E51FF004h	LDR	PC, 0x4
00000004h	48000000h	(DATA)	

2.2.1.3.2 Factory Programming

The configuration for factory programming is shown in **Figure 2-3**. Usually the Factory Programming Host connects with MT6516 via the UART interface. The download speed can be up to 921K bps while MCU is running at 26MHz.

After the system has reset, the Boot Code guides the processor to run the Factory Programming software placed in System ROM. Then, MT6516 starts and polls the UART1 port until valid information is detected.

 © 2009 MediaTek Inc.

The first information received on the UART1 is used to configure the chip for factory programming. The Flash downloader program is then transferred into System RAM or external SRAM.
Further information is detailed in the MT6516 Software Programming Specification.

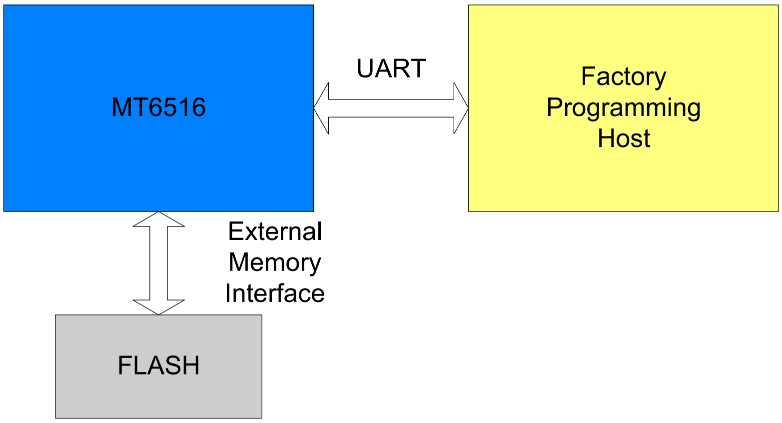


Figure 2-3 ***System configuration required for factory programming***

2.2.1.3.3 NAND Flash Booting

If MT6516 cannot receive data from UART1 for a certain amount of time, the program in System ROM checks if any valid boot loader exists in NAND flash. If found, the boot loader code is copied from NAND flash to RAM (internal or external) and executed to start the real application software. If no valid boot loader can be found in NAND flash, MT6516 starts executing code in EMI bank0 memory. The whole boot sequence is shown in the following figure.

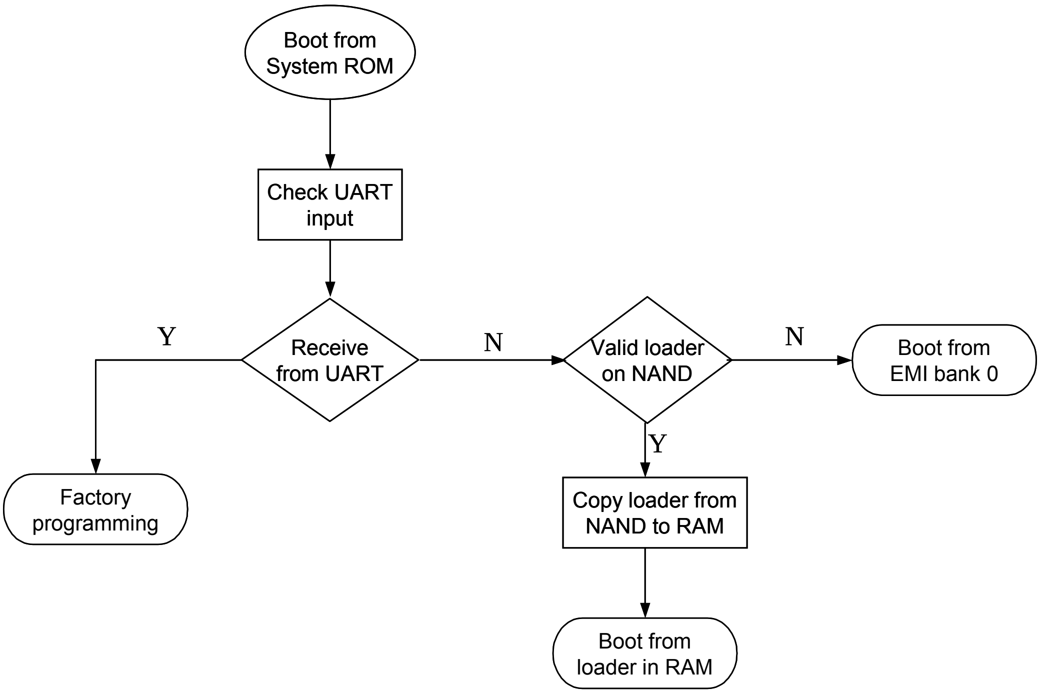


Figure 2-4 ***Boot sequence***

2.2.1.4 Little Endian Mode

The MT6516 system always treats 32-bit words of memory in Little Endian format. In Little Endian mode, the lowest numbered byte in a word is stored in the least significant position, and the highest numbered byte in the most significant position. Byte 0 of the memory system is therefore connected to data lines 7 through 0.

2.3 Bus System

2.3.1 General Description

Three levels of bus hierarchy are employed in the Micro-Controller Unit Subsystem of MT6516. As depicted in **Figure 2-1**, AHB Bus and APB Bus serve as system backbone and peripheral buses, while an APB bridge connects these two buses. Both AHB and APB Buses operate at the same or half the clock rate of processor core.

The APB Bridge is the only bus master residing on the APB bus. All APB slaves are mapped onto memory block MB8 in the MCU 32-bit addressing space. A central address decoder is implemented inside the bridge to generate select signals for individual peripherals. In addition, since the base address of each APB slave is associated with select signals, the address bus on APB contains only the value of offset address.

The maximum address space that can be allocated to a single APB slave is 64 KB, i.e. 16-bit address lines. The width of the data bus is mainly constrained to 16 bits to minimize the design complexity and power consumption while some use 32-bit data buses to accommodate more bandwidth. In the case where an APB slave needs large amount of transfers, the device driver can also request DMA channels to conduct a burst of data transfer. The base address and data width of each peripheral are listed in **Table 2-1**.

 © 2009 MediaTek Inc.

Table 2-1 ***Register Base Addresses for MCU Peripherals***

AP SIDE					
APB Brdige Definition	Address	Description	DW	Software_BASE ID	Module Name
APB BUS0 (apconfig)	8000_0000h	EFUSE	32	EFUSE_BASE	efusec
	8000_1000h	Configuration Registers (Clock, Power Down, Version and Reset)	32	CONFG_BASE	apconfg
	8000_2000h	General Purpose Inputs/Outputs	32	GPIO_BASE	gpio
	8000_3000h	Reset Generation Unit	32	RGU_BASE	rgu
APB BUS1 (apmcu)	8002_0000h	External Memory Interface	32	EMI_BASE	emi
	8002_1000h	Interrupt Controller	32	CIRQ_BASE	cirq
	8002_2000h	DMA Controller	32	DMA_BASE	dma
	8002_3000h	UART 1	16	UART1_BASE	uart
	8002_4000h	UART 2	16	UART2_BASE	uart
	8002_5000h	UART 3	16	UART3_BASE	uart
	8002_6000h	General Purpose Timer	16	GPT_BASE	apgpt
	8002_7000h	HDQ	16	HDQ_BASE	hdq_onewire
	8002_8000h	Keypad Scanner	16	KP_BASE	kp
	8002_9000h	Pulse-Width Modulation Outputs	16	PWM_BASE	pwm
	8002_B000h	UART4	16	UART4_BASE	Uart
	8002_C000h	Real Time Clock	16	RTC_BASE	rtc
	8002_D000h	SEJ	32	SEJ_BASE	sej
	8002_E000h	I2C Controller 3	16	I2C3_BASE	i2c
	8002_F000h	IrDA	16	IRDA_BASE	irda
	8003_0000h	I2C Controller 1	16	I2C_BASE	I2c
	8003_1000h	MS/SD Controller 1	32	MSDC1_BASE	msdc
	8003_2000h	NAND Flash Interface	32	NFI_BASE	nfi
	8003_3000h	SIM2	16	SIM_BASE	sim
	8003_4000h	MS/SD Controller 2	32	MSDC2_BASE	msdc
	8003_5000h	I2C Controller 2	16	I2C2_BASE	i2c
	8003_6000h	CCIF	32	CCIF_BASE	ccif
	8003_8000h	NFI ECC	32	NFIECC_BASE	nfi

© 2009 MediaTek Inc.

	8003_9000h	APMCUSYS Config	32	AMCONFG_BASE	apmcusys_confg
	8003_A000h	AP2MD Back Door	32	AP2MD_BASE	ap2md
	8003_B000h	AP Side Voice Front End	32	APVFE_BASE	vfe
	8003_C000h	AP Sleep Control	16	APSLP_BASE	ap_sleep_ctrl
	8003_D000h	AUXADC	16	AUXADC_BASE	Auxadc
	8003_E000h	AP X General Purpose Timer	16	APXGPT_BASE	apxgpt
	8003_F000h	MS/SD Controller 3	32	MSDC3_BASE	msdc
APB BUS2	8004_0000h	CoreSight Debug	32	CSDBG_BASE	csdbg
APB BUS3	8006_0000h	PLL Config	16	PLL_BASE	confg_cci
APB BUS4	8008_0000h	Graphics Memory Controller	32	GMC1_BASE	gmc1_ahb
	8008_1000h	2D Accelerator	32	G2D_BASE	g2d
	8008_2000h	2D Command Queue	32	GCMQ_BASE	gcmq
	8008_3000h	Fake Engine	32	GIFDEC_BASE	g1fake
	8008_4000h	Image DMA	32	IMGDMA_BASE	image_dma
	8008_5000h	PNG Decoder	32	PNGDEC_BASE	png_decoder
	8008_6000h				
	8008_7000h	SPI (for Mobile TV)	16	MTVSPI_BASE	spi
	8008_8000h	TV Controller	32	TVCON_BASE	tvc
	8008_9000h	TV Encoder	32	TVENC_BASE	tve
	8008_A000h	Camera Interface	32	CAM_BASE	cam
	8008_B000h	Camera ISPMEM	32	CAM_ISP_BASE	cam_ispmem
	8008_C000h	Back Light Scaling	32	BLS_BASE	Bls
	8008_D000h	Capture Resizer	32	CRZ_BASE	Crz
	8008_E000h	Drop Resizer	32	DRZ_BASE	Drz
	8008_F000h	ASM	32	ASM_BASE	asm
	8009_0000h	Wavetable	32	WT_BASE	wavetable
	8009_1000h	Image Processing	32	IMG_BASE	Imgproc
	8009_2000h	Graph1sys Config	16	GRAPH1SYS_CONFG_BASE	graph1sys_confg
APB BUS5	800A_0000h	Graphics Memory Controller	32	GMC2_BASE	gmc2
	800A_1000h	JPEG Decoder	32	JPEG_BASE	jpg
	800A_2000h	3D Engine	32	M3D_BASE	m3d
	800A_3000h	Post Processing Resizer	32	PRZ_BASE	prz
	800A_4000h	Image DMA 1	32	IMGDMA1_BASE	image_dma_1
	800A_5000h	MP4 Deblocking	32	MP4_DEBLK_BASE	mp4_deblk
	800A_6000h	Fake Engine	32	FAKE_ENG2_BASE	fake_eng_2

© 2009 MediaTek Inc.

	800A_7000h	Graph2sys Config	32	GRAPH2SYS_BASE	graph2sys_confg
APB BUS6	800C_0000h	Keypad Scanner	32	MP4_BASE	mp4
	800C_1000h	H264	32	H264_BASE	h264

2.4 UUID

2.4.1 General Description

UUID is a 128-bit codes and unique among all chips. In general, you always get the different UUIDs between any two chips. UUID can be obtained by reading the address 0x8000_0010, 0x8000_0014, 0x8000_0018, 0x8000_001c
(four 32-bit words, from LSB to MSB, total: 32*4=128 bits)

2.5 Interrupt Controller

2.5.1 General Description

Figure 2-5 outlines the major functionality of the MCU Interrupt Controller. The interrupt controller processes all interrupt sources coming from external lines and internal MCU peripherals. Since ARM926EJ-S core supports two levels of interrupt latency, this controller generates two request signals: FIQ for fast, low latency interrupt request and IRQ for more general interrupts with lower priority.

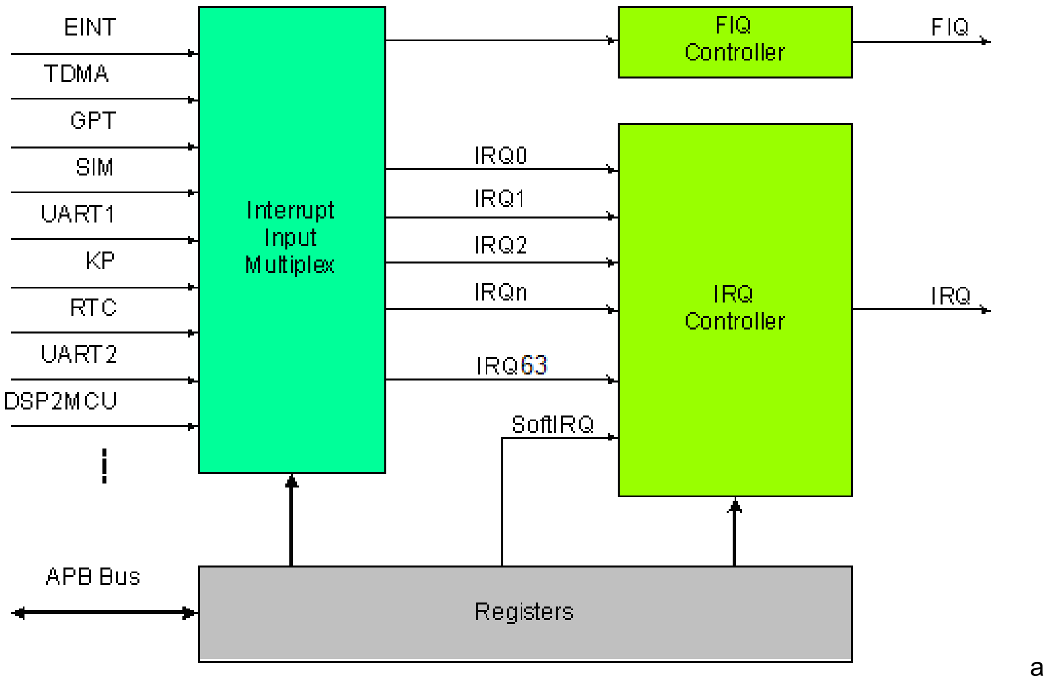


Figure 2-6 ***Block Diagram of the Interrupt Controller***

© 2009 MediaTek Inc.

One and only one of the interrupt sources can be assigned to FIQ Controller and have the highest priority in requesting timing critical service. All the others share the same IRQ signal by connecting them to IRQ Controller. The IRQ Controller manages up 64 interrupt lines of IRQ0 to IRQ63 with fixed priority in descending order.

The Interrupt Controller provides a simple software interface by mean of registers to manipulate the interrupt request shared system. IRQ Selection Registers and FIQ Selection Register determine the source priority and connecting relation among sources and interrupt lines. IRQ Source Status Register allows software program to identify the source of interrupt that generates the interrupt request. IRQ Mask Register provides software to mask out undesired sources some time. End of Interrupt Register permits software program to indicate to the controller that a certain interrupt service routine has been finished.

Binary coded version of IRQ Source Status Register is also made available for software program to helpfully identify the interrupt source. Note that while taking advantage of this feature, it should also take the binary coded version of End of Interrupt Register coincidently.

The essential Interrupt Table of ARM926EJ-S core is shown as Table 2-2.

Address	Description
00000000h	System Reset
00000018h	IRQ
0000001Ch	FIQ

Table 2-3 ***Interrupt Table of ARM926EJ-S***

2.5.1.1 Interrupt Source Masking

Interrupt controller provides the function of Interrupt Source Masking by the way of programming MASK register. Any of them can be masked individually.

However, because of the bus latency, the masking takes effect no earlier than 3 clock cycles later. In this time, the to-be-masked interrupts could come in and generate an IRQ pulse to MCU, and then disappear immediately. This IRQ forces MCU going to Interrupt Service Routine and polling Status Register (IRQ_STA(IRQ_STAH+IRQ_STAL) or IRQ_STA2), but the register shows there is no interrupt. This might cause MCU malfunction.

There are two ways for programmer to protect their software.

1. Return from ISR (Interrupt Service Routine) immediately while the Status register shows no interrupt.
2. Set I bit of MCU before doing Interrupt Masking, and then clear it after Interrupt Masking done.

Both avoid the problem, but the first item recommended to have in the ISR.

2.5.1.2 External Interrupt

This interrupt controller also integrates an External Interrupt Controller that can support up to 20 interrupt requests coming from external sources, the EINT0~19, and 4 WakeUp interrupt requests, i.e. EINT20~23, coming from peripherals. All external interrupts can inform system to resume the system clock. EINT0~4 interrupt request can be configured as from external pins or internal peripherals.

The 20 external interrupts can be used for different kind of applications, mainly for event detections: detection of hand free connection, detection of hood opening, detection of battery charger connection.

Since the external event may be unstable in a certain period, a de-bounce mechanism is introduced to ensure the functionality. The circuitry is mainly used to verify that the input signal remains stable for a programmable number of periods of the clock. When this condition is satisfied, for the appearance or the disappearance of the input, the output of the de-bounce logic changes to the desired state. Note that, because it uses the 32 KHz slow clock for performing the de-bounce process, the parameter of de-bounce period and de-bounce enable takes effect no sooner than one 32 KHz clock cycle (~31.25us) after the software program sets them. When the sources of External Interrupt Controller are used to resume the system clock in sleep mode, the de-bounce mechanism must be enabled. However, the polarities of EINTs are clocked with the system clock. However, the polarities of EINTs are clocked with the system clock. Any changes to them take effect immediately.

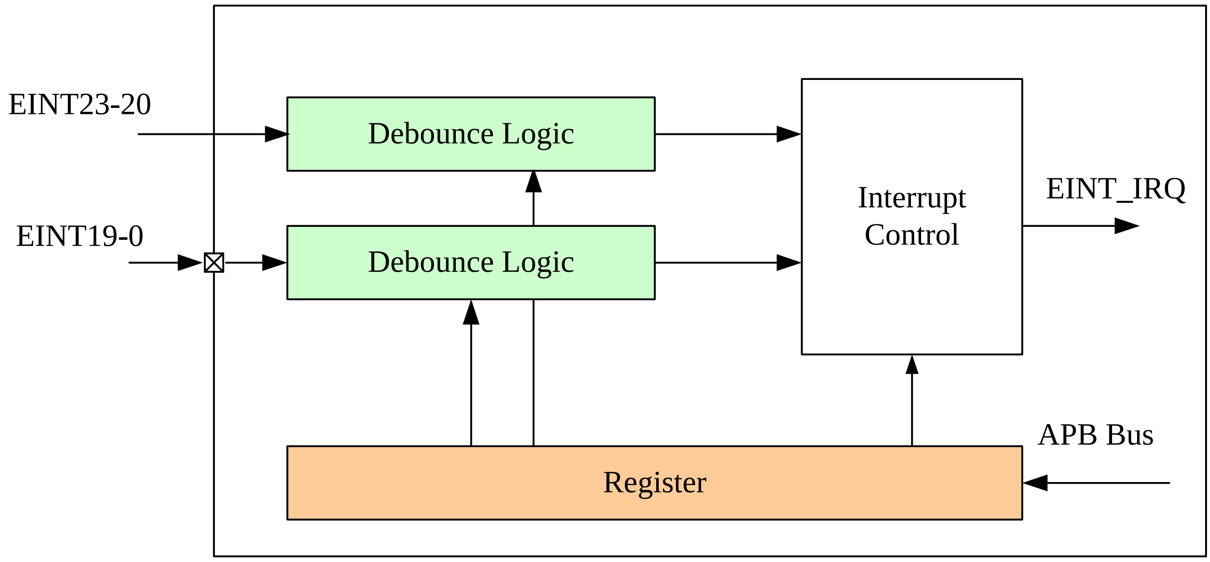


Figure 2-7 ***Block Diagram of External Interrupt Controller***

© 2009 MediaTek Inc.

2.5.1.3 External Interrupt Input Pins

EINT	Edge / Level HW Debounce	SOURCE PIN	SUPPLEMENT
EINT0	Edge / Level Yes	If (GPIO59 M==1) then EINT0= USB DP Pin else EINT0= GPIO59	1. GPIOs should be in the input mode and are effected by GPIO data input inversion registers. 2. GPIOxx_M is the GPIO mode control registers, please refer to GPIO segment.
EINT1	Edge / Level Yes	If (GPIO60 M==1) then EINT1= UART1_RXD else EINT1= GPIO60	
EINT2	Edge / Level Yes	If (GPIO61 M==1) then EINT2= URXD2 else EINT2= GPIO61	
EINT3	Edge / Level Yes	If (GPIO62 M==1) then EINT3 = URXD3 else EINT3= GPIO62	
EINT4	Edge / Level Yes	if(GPIO63_M==1) then EINT4=GPIO63 else EINT4=1	
EINT5	Edge / Level Yes	if(GPIO64_M==1) then EINT5=GPIO64 else EINT5=1	
EINT6	Edge / Level Yes	if(GPIO65_M==1) then EINT6=GPIO65 else EINT6=1	
EINT7	Edge / Level Yes	if(GPIO66_M==1) then EINT7=GPIO66 else EINT7=1	
EINT8	Edge / Level Yes	if(GPIO21_M==1) then EINT8=GPIO21 else EINT8=1	
EINT9	Edge / Level Yes	if(GPIO22_M==3) then EINT9=GPIO22 else EINT9=1	
EINT10	Edge / Level Yes	if(GPIO87_M==3) then EINT10=GPIO87 else EINT10=1	
EINT11	Edge / Level	if(GPIO88_M==3) then EINT11=GPIO88 else EINT11=1	

	Yes	
EINT12	Edge / Level Yes	if(GPIO89_M==3) then EINT12=GPIO89 else EINT12=1
EINT13	Edge / Level Yes	if(GPIO90_M==3) then EINT13=GPIO90 else EINT13=1
EINT14	Edge / Level Yes	if(GPIO91_M==3) then EINT14=GPIO91 else EINT14=1
EINT15	Edge / Level Yes	if(GPIO1_M==3) then EINT15=GPIO1 else EINT15=1
EINT16	Edge / Level Yes	if(GPIO54_M==2) then EINT16=GPIO54 else EINT16=1
EINT17	Edge / Level Yes	if(GPIO55_M==3) then EINT17=GPIO55 else EINT17=1
EINT18	Edge / Level Yes	if(GPIO56_M==2) then EINT18=GPIO56 else EINT18=1
EINT19	Edge / Level Yes	if(GPIO57_M==2) then EINT19=GPIO57 else EINT19=1
EINT20	Edge / Level Yes	USB20 IDDIG
EINT21	Edge / Level Yes	USB20 VBUSVALID
EINT22	Edge / Level Yes	CPU interface IRQ_B
EINT23	Edge / Level Yes	DSP interface IRQ_B

REGISTER	REGISTER NAME	SYNONYM

ADDRESS		
CIRQ + 0000h	IRQ Selection 0 Register	IRQ_SEL0
CIRQ + 0004h	IRQ Selection 1 Register	IRQ_SEL1
CIRQ + 0008h	IRQ Selection 2 Register	IRQ_SEL2
CIRQ + 000Ch	IRQ Selection 3 Register	IRQ_SEL3
CIRQ + 0010h	IRQ Selection 4 Register	IRQ_SEL4
CIRQ + 0014h	IRQ Selection 5 Register	IRQ_SEL5
CIRQ + 0018h	IRQ Selection 6 Register	IRQ_SEL6
CIRQ + 001ch	IRQ Selection 7 Register	IRQ_SEL7
CIRQ + 0034h	FIQ Selection Register	FIQ_SEL
CIRQ + 0038h	IRQ Mask Register (LSB)	IRQ_MASKL
CIRQ + 003ch	IRQ Mask Register (MSB)	IRQ_MASKH
CIRQ + 0040h	IRQ Mask Clear Register (LSB)	IRQ_MASK_CLRL
CIRQ + 0044h	IRQ Mask Clear Register (MSB)	IRQ_MASK_CLRH
CIRQ + 0048h	IRQ Mask Set Register (LSB)	IRQ_MASK_SETL
CIRQ + 004ch	IRQ Mask Set Register (MSB)	IRQ_MASK_SETH
CIRQ + 0050h	IRQ Status Register (LSB)	IRQ_STAL
CIRQ + 0054h	IRQ Status Register (MSB)	IRQ_STAH
CIRQ + 0058h	IRQ End of Interrupt Register (LSB)	IRQ_EOIL
CIRQ + 005ch	IRQ End of Interrupt Register (MSB)	IRQ_EOIH
CIRQ + 0060h	IRQ Sensitive Register (LSB)	IRQ_SENSL
CIRQ + 0064h	IRQ Sensitive Register (MSB)	IRQ_SENSH
CIRQ + 0068h	IRQ Software Interrupt Register (LSB)	IRQ_SOFTL
CIRQ + 006ch	IRQ Software Interrupt Register (MSB)	IRQ_SOFTH
CIRQ + 0070h	FIQ Control Register	FIQ_CON
CIRQ + 0074h	FIQ End of Interrupt Register	FIQ_EOI
CIRQ + 0078h	Binary Coded Value of IRQ_STATUS	IRQ_STA2
CIRQ + 007ch	Binary Coded Value of IRQ_EOI	IRQ_EOI2
CIRQ + 0080h	Binary Coded Value of IRQ_SOFT	IRQ_SOFT2
CIRQ + 0100h	EINT Status Register	EINT_STA
CIRQ + 0104h	EINT Mask Register	EINT_MASK
CIRQ + 0108h	EINT Mask Disable Register	EINT_MASK_DIS
CIRQ + 010Ch	EINT Mask Enable Register	EINT_MASK_EN
CIRQ + 0110h	EINT Interrupt Acknowledge Register	EINT_INTACK
CIRQ + 0114h	EINT Sensitive Register	EINT_SENS
CIRQ + 0120h	EINT0 De-bounce Control Register	EINT0_CON
CIRQ + 0130h	EINT1 De-bounce Control Register	EINT1_CON

CIRQ + 0140h	EINT2 De-bounce Control Register	EINT2_CON
CIRQ + 0150h	EINT3 De-bounce Control Register	EINT3_CON
CIRQ + 0160h	EINT4 De-bounce Control Register	EINT4_CON
CIRQ + 0170h	EINT5 De-bounce Control Register	EINT5_CON
CIRQ + 0180h	EINT6 De-bounce Control Register	EINT6_CON
CIRQ + 0190h	EINT7 De-bounce Control Register	EINT7_CON
CIRQ + 01a0h	EINT8 De-bounce Control Register	EINT8_CON
CIRQ + 01b0h	EINT9 De-bounce Control Register	EINT9_CON
CIRQ + 01c0h	EINT10 De-bounce Control Register	EINT10_CON
CIRQ + 01d0h	EINT11 De-bounce Control Register	EINT11_CON
CIRQ + 01e0h	EINT12 De-bounce Control Register	EINT12_CON
CIRQ + 01f0h	EINT13 De-bounce Control Register	EINT13_CON
CIRQ + 0200h	EINT14 De-bounce Control Register	EINT14_CON
CIRQ + 0210h	EINT15 De-bounce Control Register	EINT15_CON
CIRQ + 0220h	EINT16 De-bounce Control Register	EINT16_CON
CIRQ + 0230h	EINT17 De-bounce Control Register	EINT17_CON
CIRQ + 0240h	EINT18 De-bounce Control Register	EINT18_CON
CIRQ + 0250h	EINT19 De-bounce Control Register	EINT19_CON
CIRQ + 0260h	EINT20 De-bounce Control Register	EINT20_CON
CIRQ + 0270h	EINT21 De-bounce Control Register	EINT21_CON
CIRQ + 0280h	EINT22 De-bounce Control Register	EINT22_CON
CIRQ + 0290h	EINT23 De-bounce Control Register	EINT23_CON

Table 2-4 ***Interrupt Controller Register Map***

2.5.2 Register Definitions

CIRQ+0000h IRQ Selection 0 Register IRQ_SEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ4**						**IRQ3**						**IRQ2**	
Type			R/W						R/W						R/W	
Reset			000100b						000011b						00b	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ2**				**IRQ1**						**IRQ0**					
Type	R/W				R/W						R/W					
Reset	0010b				000001b						000000b					

CIRQ+0004h IRQ Selection 1 Register IRQ_SEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ9**						**IRQ8**						**IRQ7**	
Type			R/W						R/W						R/W	
Reset			0x9						0x8						0x7	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	IRQ7				IRQ6						IRQ5					
Type	R/W				R/W						R/W					
Reset	7				6						5					

CIRQ+0008h IRQ Selection 2 Register IRQ_SEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQE						IRQD						IRQC	
Type			R/W						R/W						R/W	
Reset			e						D						c	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQC				IRQB						IRQA					
Type	R/W				R/W						R/W					
Reset	c				b						a					

CIRQ+000ch IRQ Selection 3 Register IRQ_SEL3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQ13						IRQ12						IRQ11	
Type			R/W						R/W						R/W	
Reset			13						12						11	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ11				IRQ10						IRQF					
Type	R/W				R/W						R/W					
Reset	11				10						f					

CIRQ+0010h IRQ Selection 4 Register IRQ_SEL4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQ18						IRQ17						IRQ16	
Type			R/W						R/W						R/W	
Reset			18						17						16	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ16				IRQ15						IRQ14					
Type	R/W				R/W						R/W					
Reset	16				15						14					

CIRQ+0014h IRQ Selection 5 Register IRQ_SEL5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQ1D						IRQ1C						IRQ1B	
Type			R/W						R/W						R/W	
Reset			1d						1c						1b	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ1B				IRQ1A						IRQ19					
Type	R/W				R/W						R/W					
Reset	1b				1a						19					

CIRQ+0018h IRQ Selection 6 Register IRQ_SEL6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQ22						IRQ21						IRQ20	
Type			R/W						R/W						R/W	

 © 2009 MediaTek Inc.

Reset			22						21						20	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ20**				**IRQ1F**						**IRQ1E**					
Type	R/W				R/W						R/W					
Reset	20				1f						1e					

CIRQ+001ch IRQ Selection 7 Register IRQ_SEL7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ27**						**IRQ26**						**IRQ25**	
Type			R/W						R/W						R/W	
Reset			27						26						25	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ25**				**IRQ24**						**IRQ23**					
Type	R/W				R/W						R/W					
Reset	25				24						23					

CIRQ+0020h IRQ Selection 8 Register IRQ_SEL8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ2C**						**IRQ2B**						**IRQ2A**	
Type			R/W						R/W						R/W	
Reset			2C						2B						2A	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ2A**				**IRQ29**						**IRQ28**					
Type	R/W				R/W						R/W					
Reset	2A				29						28					

CIRQ+0024h IRQ Selection 9 Register IRQ_SEL9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ31**						**IRQ30**						**IRQ2F**	
Type			R/W						R/W						R/W	
Reset			31						30						2F	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ2F**				**IRQ2E**						**IRQ2D**					
Type	R/W				R/W						R/W					
Reset	2F				2E						2D					

CIRQ+0028h IRQ Selection 10 Register IRQ_SEL10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ36**						**IRQ35**						**IRQ34**	
Type			R/W						R/W						R/W	
Reset			36						35						34	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ34**				**IRQ33**						**IRQ32**					
Type	R/W				R/W						R/W					
Reset	34				33						32					

CIRQ+002Ch IRQ Selection 11 Register IRQ_SEL11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name			**IRQ3B**						**IRQ3A**						**IRQ39**	
Type			R/W						R/W						R/W	
Reset			3B						3A						39	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ39**				**IRQ38**						**IRQ37**					
Type	R/W				R/W						R/W					
Reset	39				38						37					

CIRQ+0030h IRQ Selection 12 Register IRQ_SEL12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**IRQ3F**						**IRQ3E**	
Type									R/W						R/W	
Reset									3F						3E	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ3E**				**IRQ3D**						**IRQ3C**					
Type	R/W				R/W						R/W					
Reset	3E				3D						3C					

CIRQ+0034h FIQ Selection Register FIQ_SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**FIQ**					
Type											R/W					
Reset											0					

The IRQ/FIQ Selection Registers provide system designers with a flexible routing scheme to make various mappings of priority among interrupt sources possible. The registers allow the interrupt sources to be mapped onto interrupt requests of either FIQ or IRQ. While only one interrupt source can be assigned to FIQ, the other ones share IRQs by mapping them onto IRQ0 to IRQ3F connected to IRQ controller. The priority sequence of IRQ0~IRQ3F is fixed, i.e. IRQ0 > IRQ1 > IRQ2 > … > IRQ3E > IRQ3F. During the software configuration process, the Interrupt Source Code of desired interrupt source should be written into source field of the corresponding IRQ_SEL0-IRQ_SEL12/FIQ_SEL. 6-bit Interrupt Source Codes for all interrupt sources are fixed and defined.

Interrupt Source	STA2 (Hex)	STAH_STAL
GPI_FIQ	0	0000_00000001
SIM2	1	0000_00000002
DMA	2	0000_00000004
UART1	3	0000_00000008
KP	4	0000_00000010
UART2	5	0000_00000020
GPT	6	0000_00000040
EINT	7	0000_00000080

© 2009 MediaTek Inc.

USB	8	0000_00000100
RTC	9	0000_00000200
MSDC1	a	0000_00000400
IRDA	b	0000_00000800
LCD	c	0000_00001000
UART3	d	0000_00002000
GPI	e	0000_00004000
WDT	f	0000_00008000
TVC	10	0000_00010000
I2C3	11	0000_00020000
NFI	12	0000_00040000
I2C2	13	0000_00080000
Image DMA	14	0000_00100000
Image DMA2	15	0000_00200000
PNG	16	0000_00400000
I2C	17	0000_00800000
G2D	18	0000_01000000
Image Proc	19	0000_02000000
CAM	1a	0000_04000000
MPEG4 Decode	1b	0000_08000000
MPEG4 Encode	1c	0000_10000000
JPEG Decode	1d	0000_20000000
JPEG Encode	1e	0000_40000000
Resizer CRZ	1f	0000_80000000
Resizer DRZ	20	0001_00000000
Resizer PRZ	21	0002_00000000
TVE	22	0004_00000000
USB DMA	23	0008_00000000
PWM	24	0010_00000000
MPEG4 Deblock	25	0020_00000000
H264 Decode	26	0040_00000000
MSDC1 Event	27	0080_00000000
DPI	28	0100_00000000
APCCIF	29	0200_00000000
M3D	2a	0400_00000000
EMI	2b	0800_00000000
MSDC2	2c	1000_00000000
MSDC2 Event	2d	2000_00000000

 © 2009 MediaTek Inc.

Reserved	2e	4000_00000000
CEVA CCIF	2f	8000_00000000
NFI ECC	30	1_0000_00000000
WAVETABLE	31	2_0000_00000000
DVF Controller	32	4_0000_00000000
Reserved	33	8_0000_00000000
GMC1	34	10_0000_00000000
GMC2	35	20_0000_00000000
AP_SLEEP_CTRL	36	40_0000_00000000
ASM	37	80_0000_00000000
Touch Screen	38	100_0000_00000000
APXGPT	39	200_0000_00000000
LOWBAT	3a	400_0000_00000000
Mobile TV SPI	3b	800_0000_00000000
UART4	3c	1000_0000_00000000
MSDC3	3d	2000_0000_00000000
MSDC3 Event	3e	4000_0000_00000000
OneWire	3f	8000_0000_00000000

Table 2-5 ***Interrupt Source Code for Interrupt Sources***

- **FIQ, IRQ0-26** The 6-bit content of this field corresponds to an Interrupt Source Code shown above.

CIRQ+0038h IRQ Mask Register (LSB) IRQ_MASKL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CIRQ+003ch IRQ Mask Register (MSB) IRQ_MASKH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name	**IRQ3F**	**IRQ3E**	**IRQ3D**	**IRQ3C**	**IRQ3B**	**IRQ3A**	**IRQ39**	**IRQ38**	**IRQ37**	**IRQ36**	**IRQ35**	**IRQ34**	**IRQ33**	**IRQ32**	**IRQ31**	**IRQ30**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ2F**	**IRQ2E**	**IRQ2D**	**IRQ2C**	**IRQ2B**	**IR2A**	**IRQ29**	**IRQ28**	**IRQ27**	**IRQ26**	**IRQ25**	**IRQ24**	**IRQ23**	**IRQ22**	**IRQ21**	**IRQ20**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register contains a mask bit for each interrupt line in IRQ Controller. The register allows each interrupt source IRQ0 to IRQ1F to be disabled or masked separately under software control. After a system reset, all bit values are set to 1 to indicate that interrupt requests are prohibited.

- **IRQ0-3F** Mask control for the associated interrupt source in the IRQ controller
  - **0** Interrupt is enabled.
  - **1** Interrupt is disabled.

CIRQ+0040h IRQ Mask Clear Register (LSB) IRQ_MASK_CLRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ1F**	**IRQ1E**	**IRQ1D**	**IRQ1C**	**IRQ1B**	**IRQ1A**	**IRQ19**	**IRQ18**	**IRQ17**	**IRQ16**	**IRQ15**	**IRQ14**	**IRQ13**	**IRQ12**	**IRQ11**	**IRQ10**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQF**	**IRQE**	**IRQD**	**IRQC**	**IRQB**	**IRQA**	**IRQ9**	**IRQ8**	**IRQ7**	**IRQ6**	**IRQ5**	**IRQ4**	**IRQ3**	**IRQ2**	**IRQ1**	**IRQ0**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

CIRQ+0044h IRQ Mask Clear Register (MSB) IRQ_MASK_CLRH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ3F**	**IRQ3E**	**IRQ3D**	**IRQ3C**	**IRQ3B**	**IRQ3A**	**IRQ39**	**IRQ38**	**IRQ37**	**IRQ36**	**IRQ35**	**IRQ34**	**IRQ33**	**IRQ32**	**IRQ31**	**IRQ30**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQ2F**	**IRQ2E**	**IRQ2D**	**IRQ2C**	**IRQ2B**	**IR2A**	**IRQ29**	**IRQ28**	**IRQ27**	**IRQ26**	**IRQ25**	**IRQ24**	**IRQ23**	**IRQ22**	**IRQ21**	**IRQ20**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

This register is used to clear bits in IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be cleared. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

- **IRQ0-3F** Clear corresponding bits in IRQ Mask Register.
  - **0** No effect.
  - **1** Disable the corresponding MASK bit.

CIRQ+0048h IRQ Mask SET Register (LSB) IRQ_MASK_SETL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ1F**	**IRQ1E**	**IRQ1D**	**IRQ1C**	**IRQ1B**	**IRQ1A**	**IRQ19**	**IRQ18**	**IRQ17**	**IRQ16**	**IRQ15**	**IRQ14**	**IRQ13**	**IRQ12**	**IRQ11**	**IRQ10**
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

© 2009 MediaTek Inc.

Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S

CIRQ+004ch IRQ Mask SET Register (MSB) IRQ_MASK_SETH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ3F	IRQ3E	IRQ3D	IRQ3C	IRQ3B	IRQ3A	IRQ39	IRQ38	IRQ37	IRQ36	IRQ35	IRQ34	IRQ33	IRQ32	IRQ31	IRQ30
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ2F	IRQ2E	IRQ2D	IRQ2C	IRQ2B	IR2A	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ20
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S

This register is used to set bits in the IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be set. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

- **IRQ0-3F** Set corresponding bits in IRQ Mask Register.
  - **0** No effect.
  - **1** Enable corresponding MASK bit.

CIRQ+0050h IRQ Source Status Register (LSB) IRQ_STAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CIRQ+0054h IRQ Source Status Register (MSB) IRQ_STAH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ3F	IRQ3E	IRQ3D	IRQ3C	IRQ3B	IRQ3A	IRQ39	IRQ38	IRQ37	IRQ36	IRQ35	IRQ34	IRQ33	IRQ32	IRQ31	IRQ30
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ2F	IRQ2E	IRQ2D	IRQ2C	IRQ2B	IR2A	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ20
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This Register allows software to poll which interrupt line has generated an IRQ interrupt request. A bit set to 1 indicates a corresponding active interrupt line. Only one flag is active at a time. The IRQ_STA is type of read-clear; write access has no effect on the content.

- **IRQ0-3F** Interrupt indicator for the associated interrupt source.
  - **0** The associated interrupt source is non-active.
  - **1** The associated interrupt source is asserted.

 © 2009 MediaTek Inc.

CIRQ+0058h IRQ End of Interrupt Register (LSB) IRQ_EOIL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CIRQ+005ch IRQ End of Interrupt Register (MSB) IRQ_EOIH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ3F	IRQ3E	IRQ3D	IRQ3C	IRQ3B	IRQ3A	IRQ39	IRQ38	IRQ37	IRQ36	IRQ35	IRQ34	IRQ33	IRQ32	IRQ31	IRQ30
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ2F	IRQ2E	IRQ2D	IRQ2C	IRQ2B	IR2A	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ20
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register provides a mean for software to relinquish and to refresh the interrupt controller. Writing a 1 to a specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

- **IRQ0-3F** End of Interrupt command for the associated interrupt line.
  - **0** No service is currently in progress or pending.
  - **1** Interrupt request is in-service.

CIRQ+0060h IRQ Sensitive Register (LSB) IRQ_SENSL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CIRQ+0064h IRQ Sensitive Register (MSB) IRQ_SENSH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ3F	IRQ3E	IRQ3D	IRQ3C	IRQ3B	IRQ3A	IRQ39	IRQ38	IRQ37	IRQ36	IRQ35	IRQ34	IRQ33	IRQ32	IRQ31	IRQ30
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ2F	IRQ2E	IRQ2D	IRQ2C	IRQ2B	IR2A	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ20
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All interrupt lines of IRQ Controller, IRQ0~IRQ1F can be programmed as either edge or level sensitive. By default, all the interrupt lines are edge sensitive and should be active LOW. Once a interrupt line is

© 2009 MediaTek Inc.

programmed as edge sensitive, an interrupt request is triggered only at the falling edge of interrupt line, and the next interrupt is not accepted until the EOI command is given. However, level sensitive interrupts trigger is according to the signal level of the interrupt line. Once the interrupt line become from HIGH to LOW, an interrupt request is triggered, and another interrupt request is triggered if the signal level remain LOW after an EOI command. Note that in edge sensitive mode, even if the signal level remains LOW after EOI command, another interrupt request is not triggered. That is because edge sensitive interrupt is only triggered at the falling edge.

- **IRQ0-3F** Sensitivity type of the associated Interrupt Source
  - **0** Edge sensitivity with active LOW
  - **1** Level sensitivity with active LOW

CIRQ+0068h IRQ Software Interrupt Register (LSB) IRQ_SOFTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CIRQ+006ch IRQ Software Interrupt Register (MSB) IRQ_SOFTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ3F	IRQ3E	IRQ3D	IRQ3C	IRQ3B	IRQ3A	IRQ39	IRQ38	IRQ37	IRQ36	IRQ35	IRQ34	IRQ33	IRQ32	IRQ31	IRQ30
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ2F	IRQ2E	IRQ2D	IRQ2C	IRQ2B	IR2A	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ20
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting “1” to the specific bit position generates a software interrupt for corresponding interrupt line before mask. This register is used for debug purpose.

- **IRQ0-IRQ3F** Software Interrupt

CIRQ+0070h FIQ Control Register FIQ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SENS	MASK
Type															R/W	R/W
Reset															0	1

This register provides a means for software program to control the FIQ controller.

- **MASK** Mask control for the FIQ Interrupt Source

- **0** Interrupt is enabled.
- **1** Interrupt is disabled.
- **SENS** Sensitivity type of the FIQ Interrupt Source
  - **0** Edge sensitivity with active LOW
  - **1** Level sensitivity with active LOW

CIRQ+0074h FIQ End of Interrupt Register FIQ_EOI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**EOI**
Type																WO
Reset																0

This register provides a means for software to relinquish and to refresh the FIQ controller. Writing a '1' to the specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

- **EOI** End of Interrupt command

CIRQ+0078h Binary Coded Value of IRQ_STATUS IRQ_STA2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**NOIRQ**			**STS**					
Type								RC			RC					
Reset								0			0					

This Register is a binary coded version of IRQ_STA. It is used by the software program to poll which interrupt line has generated the IRQ interrupt request in a much easier way. Any read to it has the same result as reading IRQ_STA. The IRQ_STA2 is also read-only and read-clear; write access has no effect on the content. Note that IRQ_STA2 should be coupled with IRQ_EOI2 while using it.

- **STS** Binary coded value of IRQ_STA
- **NOIRQ** Indicating if there is an IRQ or not. If there is no IRQ, this bit is HIGH, and the value of STS is 00_0000b.

CIRQ+007ch Binary Coded Value of IRQ_EOI IRQ_EOI2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**EOI**					
Type											WO					
Reset											0					

This register is a binary coded version of IRQ_EOI. It provides an easier way for software program to relinquish and to refresh the interrupt controller. Writing a specific code results in an End of Interrupt command issued internally to the corresponding interrupt line. Note that IRQ_EOI2 should be coupled with IRQ_STA2 while using it.

- **EOI** Binary coded value of IRQ_EOI

CIRQ+0080h Binary Coded Value of IRQ_SOFT IRQ_SOFT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											SOFT					
Type											WO					
Reset											0					

This register is a binary coded version of IRQ_SOFT.

- **SOFT** Binary Coded Value of IRQ_SOFT

CIRQ+0100h EINT Interrupt Status Register EINT_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									RO	RO	RO	RO	RO	RO	RO	RO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register keeps up with current status that which EINT Source generates the interrupt request. The status will be changed to zero if the corresponding EINT source mask bit is set.

- **EINT0-EINT23** Interrupt status
  - **0** No interrupt request is generated.
  - **1** Interrupt request is pending.

CIRQ+0104h EINT Interrupt Mask Register EINT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

 © 2009 MediaTek Inc.

This register controls whether or not EINT Source is allowed to generate an interrupt request. Setting a "1" to the specific bit position prohibits the external interrupt line from becoming active.

- **EINT0-EINT23** Interrupt Mask
  - **0** Interrupt request is enabled.
  - **1** Interrupt request is disabled.

CIRQ+0108h EINT Interrupt Mask Clear Register EINT_MASK_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

This register is used to clear individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are cleared (set to 0). Otherwise the interrupt mask bit retains its original value.

- **EINT0-EINT23** Disable mask for the associated external interrupt source.
  - **0** No effect.
  - **1** Disable the corresponding MASK bit.

CIRQ+010Ch EINT Interrupt Mask Set Register EINT_MASK_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S

This register is used to set individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are set to 1. Otherwise the interrupt mask bit retains its original value.

- **EINT0-EINT23** Disable mask for the associated external interrupt source.
  - **0** No effect.
  - **1** Enable corresponding MASK bit.

CIRQ+0110h EINT Interrupt Acknowledge Register EINT_INTACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									WO	WO	WO	WO	WO	WO	WO	WO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Writing "1" to the specific bit position is to acknowledge the interrupt request that correspondingly to the external interrupt line source. Write this register to clear edge sensitive EINT triggered status. Write this register to clear EINT edge status first, if the EINT source is changed from level sensitive to edge sensitive.

- **EINT0-EINT23** Interrupt acknowledgement
  - **0** No effect
  - **1** Interrupt request is acknowledged.

CIRQ+0114h EINT Sensitive Register EINT_SENS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Sensitivity type of external interrupt source.

- **EINT0-EINT23** Sensitivity type of the associated external interrupt source.
  - **0** Edge sensitivity
  - **1** Level sensitivity

CIRQ+0120h+ n*10h EINTn De-bounce Control Register EINTn_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN				POL	CNT										
Type	R/W				R/W	R/W										
Reset	0				0	0										

These registers control the de-bounce logic for external interrupt sources in order to minimize the possibility of false activations. Note that n is from 0 to 23

When the external interrupt sources is used to resume the system clock from the sleep mode, the De-bounce control circuit must be enabled.

- **CNT** De-bounce duration in terms of number of 32 KHz clock cycles.
- **POL** Activation type of the EINT source
  - **0** Negative polarity
  - **1** Positive polarity
- **EN** De-bounce control circuit
  - **0** Disable

- **1** Enable

2.6 Direct Memory Access

2.6.1 General Description

A generic DMA Controller is placed on Layer 2 AHB Bus to support fast data transfers and to off-load the processor. With this controller, specific devices on AHB or APB buses can benefit greatly from quick completion of data movement from or to memory modules such as Internal System RAM or External SRAM, excluding TCM. TCM is invisible for DMA engine. Such Generic DMA Controller can also be used to connect any two devices other than memory module as long as they can be addressed in memory space.

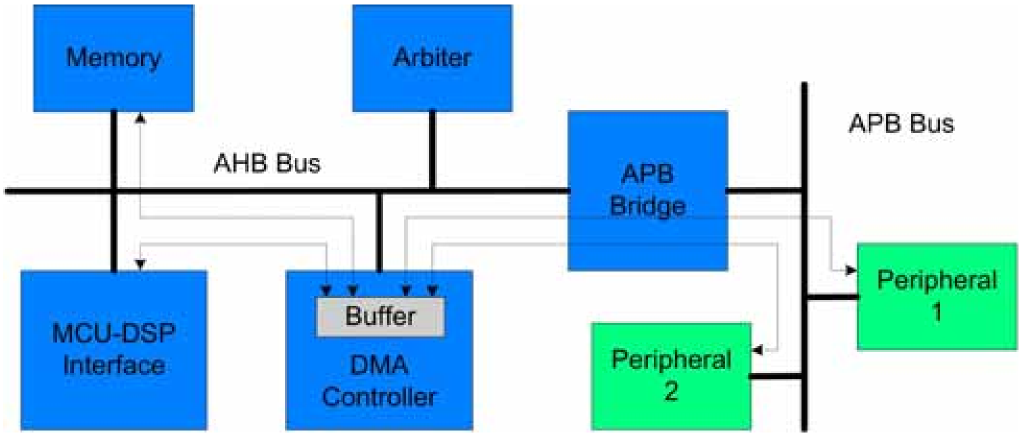


Figure 8 ***Varity Data Paths of DMA Transfers***

Up to 24 channels of simultaneous data transfers are supported. They are channel 1 to channel 24. Each channel has a similar set of registers to be configured to different scheme as desired. If more than 24 devices are requesting the DMA resources at the same time, software based arbitration should be employed. Once the service candidate is decided, the responsible device driver should configure the Generic DMA Controller properly in order to conduct DMA transfers. Both Interrupt and Polling based schemes in handling the completion event are supported. The block diagram of such generic DMA Controller is illustrated in **Figure 9**.

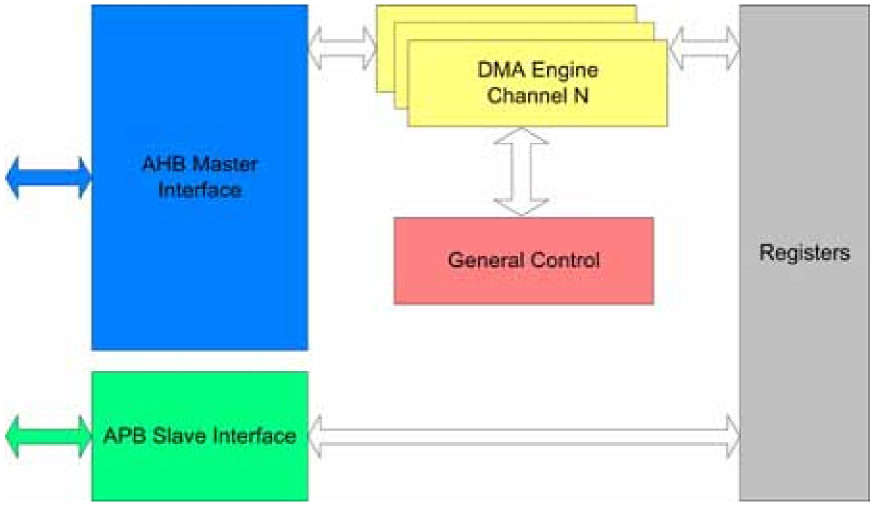


Figure 9 ***Block Diagram of Direct Memory Access Module***

2.6.1.1 Full-Size , Half-Size & Virtual FIFO DMA Channels

There are three types of DMA channels in the DMA controller. The first one is called a full-size DMA channel, the second one is called a half-size DMA channel, and the last is Virtual FIFO DMA. Channels 1 through 8 are full-size DMA channels; channels 9 through 16 are half-size ones; and channels 17 through 24 are Virtual FIFO DMA channels. The difference between the first two types of DMA channels is that both source and destination address are programmable in full-size DMA channels, but only the address of one side can be programmed in half-size DMA channel. In half-size channels, only either the source or destination address can be programmed, while the addresses of the other side is preset. Which preset address is used depends on the setting of MAS in DMA Channel Control Register. Refer to the Register Definition section for more detail.

2.6.1.2 Ring Buffer & Double Buffer Memory Data Movement

DMA channels 1 through 16 support ring-buffer and double-buffer memory data movement. This can be achieved by programming DMA_WPPT and DMA_WPTO, as well as setting WPEN in DMA_CON register to enable. **Figure 10** illustrates how this function works. Once the transfer counter reaches the value of WPPT, the next address jumps to the WPTO address after completing the WPPT data transfer. Note that only one side can be configured as ring-buffer or double-buffer memory, and this is controlled by WPSD in DMA_CON register.

 © 2009 MediaTek Inc.

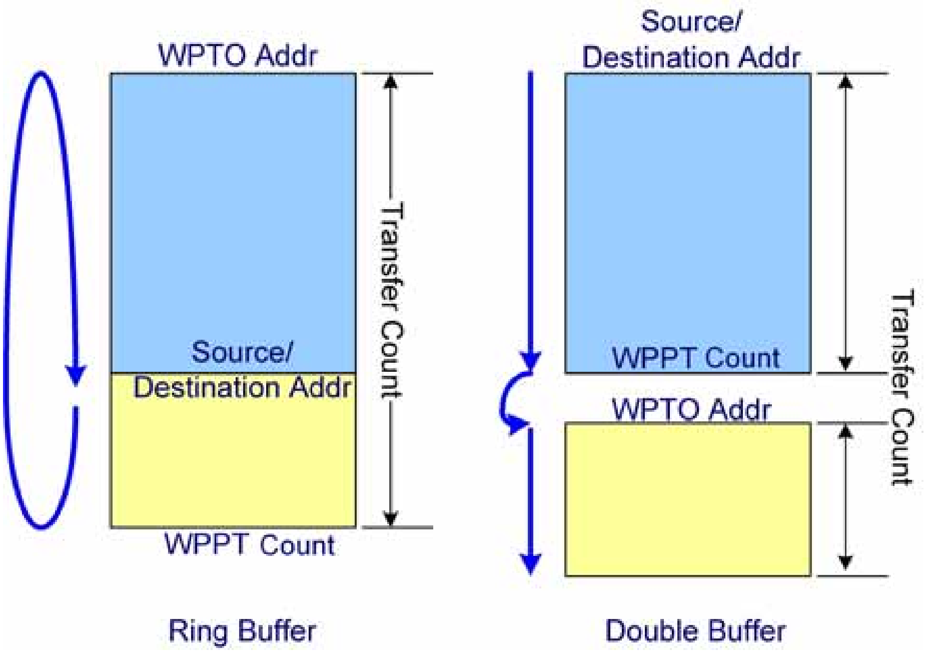


Figure 10 ***Ring Buffer and Double Buffer Memory Data Movement***

Unaligned Word Access

The address of word access on AHB bus must be aligned to word boundary, or the 2 LSB is truncated to 00b. If programmers do not notice this, it may cause an incorrect data fetch. In the case where data is to be moved from unaligned addresses to aligned addresses, the word is usually first split into four bytes and then moved byte by byte. This result in four read and four write transfers on the bus.

To improve bus efficiency, unaligned-word access is provided in DMA9~16. While this function is enabled, DMAs move data from unaligned address to aligned address by executing four continuous byte-read access and one word-write access, reducing the number of transfers on the bus by three.

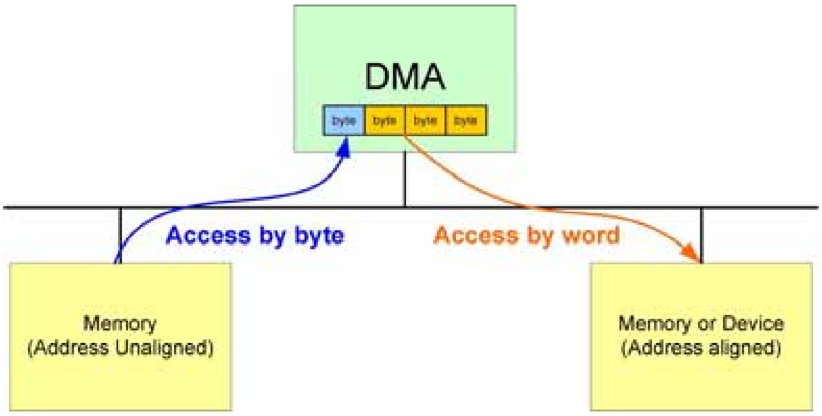


Figure 11 ***Unaligned Word Access***

2.6.1.3 Virtual FIFO DMA

Virtual FIFO DMA is used to ease UART control. The difference between the Virtual FIFO DMAs and the ordinary DMAs is that Virtual FIFO DMA contains additional FIFO controller. The read and write pointers are kept in the Virtual FIFO DMA. During a read from the FIFO, the read pointer points to the address of the next data. During a write to the FIFO, the write pointer moves to the next address. If the FIFO is empty, a FIFO read is not allowed. Similarly, data is not written into the FIFO if the FIFO is full. Due to UART flow control requirements, an alert length is programmed. Once the FIFO Space is less than this value, an alert signal is issued to enable UART flow control. The type of flow control performed depends on the setting in UART.

Each Virtual FIFO DMA can be programmed as RX or TX FIFO. This depends on the setting of DIR in DMA_CON register. If DIR is “0”(READ), it means TX FIFO. On the other hand, if DIR is “1”(WRITE), the Virtual FIFO DMA is specified as a RX FIFO.

Virtual FIFO DMA provides an interrupt to MCU. This interrupt informs MCU that there is data in the FIFO, and the amount of data is over or under the value defined in DMA_COUNT register. With this, MCU does not need to poll DMA to know when data must be removed from or put into the FIFO.

Note that Virtual FIFO DMAs cannot be used as generic DMAs, i.e. DMA1~16.

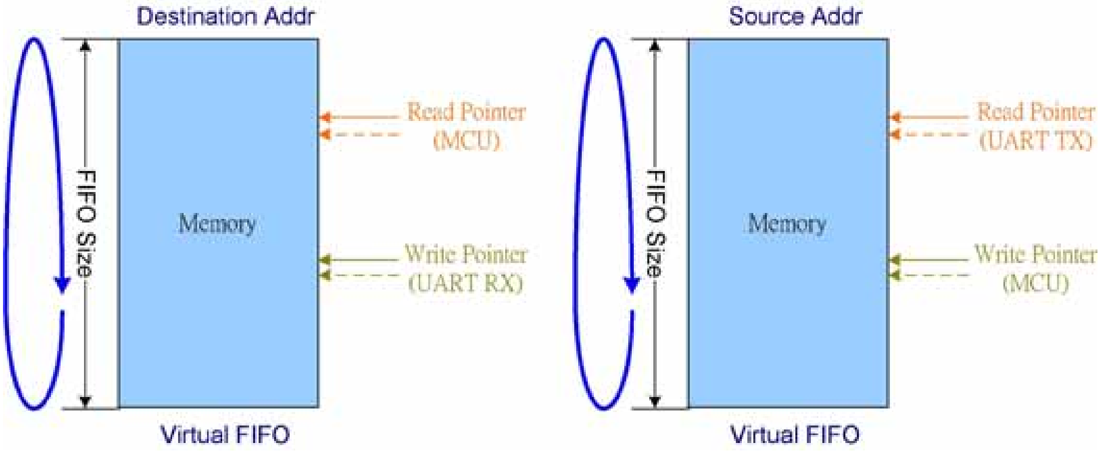


Figure 12 ***Virtual FIFO DMA***

DMA number	Address of Virtual FIFO Access Port
DMA17	8011_0000h
DMA18	8011_0100h
DMA19	8011_0200h
DMA20	8011_0300h
DMA21	8011_0400h
DMA22	8011_0500h
DMA23	8011_0600h
DMA24	8011_0700h

Table 6 Virtual FIFO Access Port

DMA number	Type	Ring Buffer	Double Buffer	Burst Mode	Unaligned Word Access
DMA1	Full Size	●	●	●	
DMA2	Full Size	●	●	●	
DMA3	Full Size	●	●	●	
DMA4	Full Size	●	●	●	
DMA5	Full Size	●	●	●	
DMA6	Full Size	●	●	●	
DMA7	Full Size	●	●	●	
DMA8	Full Size	●	●	●	
DMA9	Half Size	●	●	●	●
DMA10	Half Size	●	●	●	●
DMA11	Half Size	●	●	●	●
DMA12	Half Size	●	●	●	●
DMA13	Half Size	●	●	●	●
DMA14	Half Size	●	●	●	●
DMA15	Half Size	●	●	●	●
DMA16	Half Size	●	●	●	●
DMA17	Virtual FIFO	●			
DMA18	Virtual FIFO	●			
DMA19	Virtual FIFO	●			
DMA20	Virtual FIFO	●			
DMA21	Virtual FIFO	●			
DMA22	Virtual FIFO	●			
DMA23	Virtual FIFO	●			
DMA24	Virtual FIFO	●			

Table 7 Function List of DMA channels

REGISTER ADDRESS	REGISTER NAME	SYNONYM
DMA + 0000h	DMA Global Status Register	DMA_GLBSTA
DMA + 0004h	DMA Global Status 2 Register	DMA_GLBSTA2
DMA + 0028h	DMA Global Bandwidth Limiter Register	DMA_GLBLIMITER
DMA + 0080h	DMA Channel 1 Source Address Register	DMA1_SRC
DMA + 0084h	DMA Channel 1 Destination Address Register	DMA1_DST
DMA + 0088h	DMA Channel 1 Wrap Point Address Register	DMA1_WPPT
DMA + 008Ch	DMA Channel 1 Wrap To Address Register	DMA1_WPTO

DMA + 0090h	DMA Channel 1 Transfer Count Register	DMA1_COUNT
DMA + 0094h	DMA Channel 1 Control Register	DMA1_CON
DMA + 0098h	DMA Channel 1 Start Register	DMA1_START
DMA + 009Ch	DMA Channel 1 Interrupt Status Register	DMA1_INTSTA
DMA + 00A0h	DMA Channel 1 Interrupt Acknowledge Register	DMA1_ACKINT
DMA + 00A4h	DMA Channel 1 Remaining Length of Current Transfer	DMA1_RLCT
DMA + 00A8h	DMA Channel 1 Bandwidth Limiter Register	DMA1_LIMITER
DMA + 0100h	DMA Channel 2 Source Address Register	DMA2_SRC
DMA + 0104h	DMA Channel 2 Destination Address Register	DMA2_DST
DMA + 0108h	DMA Channel 2 Wrap Point Address Register	DMA2_WPPT
DMA + 010Ch	DMA Channel 2 Wrap To Address Register	DMA2_WPTO
DMA + 0110h	DMA Channel 2 Transfer Count Register	DMA2_COUNT
DMA + 0114h	DMA Channel 2 Control Register	DMA2_CON
DMA + 0118h	DMA Channel 2 Start Register	DMA2_START
DMA + 011Ch	DMA Channel 2 Interrupt Status Register	DMA2_INTSTA
DMA + 0120h	DMA Channel 2 Interrupt Acknowledge Register	DMA2_ACKINT
DMA + 0124h	DMA Channel 2 Remaining Length of Current Transfer	DMA2_RLCT
DMA + 0128h	DMA Channel 2 Bandwidth Limiter Register	DMA2_LIMITER
DMA + 0180h	DMA Channel 3 Source Address Register	DMA3_SRC
DMA + 0184h	DMA Channel 3 Destination Address Register	DMA3_DST
DMA + 0188h	DMA Channel 3 Wrap Point Address Register	DMA3_WPPT
DMA + 018Ch	DMA Channel 3 Wrap To Address Register	DMA3_WPTO
DMA + 0190h	DMA Channel 3 Transfer Count Register	DMA3_COUNT
DMA + 0194h	DMA Channel 3 Control Register	DMA3_CON
DMA + 0198h	DMA Channel 3 Start Register	DMA3_START
DMA + 019Ch	DMA Channel 3 Interrupt Status Register	DMA3_INTSTA
DMA + 01A0h	DMA Channel 3 Interrupt Acknowledge Register	DMA3_ACKINT
DMA + 01A4h	DMA Channel 3 Remaining Length of Current Transfer	DMA3_RLCT
DMA + 01A8h	DMA Channel 3 Bandwidth Limiter Register	DMA3_LIMITER
DMA + 0200h	DMA Channel 4 Source Address Register	DMA4_SRC
DMA + 0204h	DMA Channel 4 Destination Address Register	DMA4_DST
DMA + 0208h	DMA Channel 4 Wrap Point Address Register	DMA4_WPPT
DMA + 020Ch	DMA Channel 4 Wrap To Address Register	DMA4_WPTO
DMA + 0210h	DMA Channel 4 Transfer Count Register	DMA4_COUNT
DMA + 0214h	DMA Channel 4 Control Register	DMA4_CON

© 2009 MediaTek Inc.

DMA + 0218h	DMA Channel 4 Start Register	DMA4_START
DMA + 021Ch	DMA Channel 4 Interrupt Status Register	DMA4_INTSTA
DMA + 0220h	DMA Channel 4 Interrupt Acknowledge Register	DMA4_ACKINT
DMA + 0224h	DMA Channel 4 Remaining Length of Current Transfer	DMA4_RLCT
DMA + 0228h	DMA Channel 4 Bandwidth Limiter Register	DMA4_LIMITER
DMA + 0280h	DMA Channel 5 Source Address Register	DMA5_SRC
DMA + 0284h	DMA Channel 5 Destination Address Register	DMA5_DST
DMA + 0288h	DMA Channel 5 Wrap Point Address Register	DMA5_WPPT
DMA + 028Ch	DMA Channel 5 Wrap To Address Register	DMA5_WPTO
DMA + 0290h	DMA Channel 5 Transfer Count Register	DMA5_COUNT
DMA + 0294h	DMA Channel 5 Control Register	DMA5_CON
DMA + 0298h	DMA Channel 5 Start Register	DMA5_START
DMA + 029Ch	DMA Channel 5 Interrupt Status Register	DMA5_INTSTA
DMA + 02A0h	DMA Channel 5 Interrupt Acknowledge Register	DMA5_ACKINT
DMA + 02A5h	DMA Channel 5 Remaining Length of Current Transfer	DMA5_RLCT
DMA + 02A8h	DMA Channel 5 Bandwidth Limiter Register	DMA5_LIMITER
DMA + 0300h	DMA Channel 6 Source Address Register	DMA6_SRC
DMA + 0304h	DMA Channel 6 Destination Address Register	DMA6_DST
DMA + 0308h	DMA Channel 6 Wrap Point Address Register	DMA6_WPPT
DMA + 030Ch	DMA Channel 6 Wrap To Address Register	DMA6_WPTO
DMA + 0310h	DMA Channel 6 Transfer Count Register	DMA6_COUNT
DMA + 0314h	DMA Channel 6 Control Register	DMA6_CON
DMA + 0318h	DMA Channel 6 Start Register	DMA6_START
DMA + 031Ch	DMA Channel 6 Interrupt Status Register	DMA6_INTSTA
DMA + 0320h	DMA Channel 6 Interrupt Acknowledge Register	DMA6_ACKINT
DMA + 0324h	DMA Channel 6 Remaining Length of Current Transfer	DMA6_RLCT
DMA + 0328h	DMA Channel 6 Bandwidth Limiter Register	DMA6_LIMITER
DMA + 0380h	DMA Channel 7 Source Address Register	DMA7_SRC
DMA + 0384h	DMA Channel 7 Destination Address Register	DMA7_DST
DMA + 0388h	DMA Channel 7 Wrap Point Address Register	DMA7_WPPT
DMA + 038Ch	DMA Channel 7 Wrap To Address Register	DMA7_WPTO
DMA + 0390h	DMA Channel 7 Transfer Count Register	DMA7_COUNT
DMA + 0394h	DMA Channel 7 Control Register	DMA7_CON
DMA + 0398h	DMA Channel 7 Start Register	DMA7_START
DMA + 039Ch	DMA Channel 7 Interrupt Status Register	DMA7_INTSTA

DMA + 03A0h	DMA Channel 7 Interrupt Acknowledge Register	DMA7_ACKINT
DMA + 03A4h	DMA Channel 7 Remaining Length of Current Transfer	DMA7_RLCT
DMA + 03A8h	DMA Channel 7 Bandwidth Limiter Register	DMA7_LIMITER
DMA + 0400h	DMA Channel 8 Source Address Register	DMA8_SRC
DMA + 0404h	DMA Channel 8 Destination Address Register	DMA8_DST
DMA + 0408h	DMA Channel 8 Wrap Point Address Register	DMA8_WPPT
DMA + 040Ch	DMA Channel 8 Wrap To Address Register	DMA8_WPTO
DMA + 0410h	DMA Channel 8 Transfer Count Register	DMA8_COUNT
DMA + 0414h	DMA Channel 8 Control Register	DMA8_CON
DMA + 0418h	DMA Channel 8 Start Register	DMA8_START
DMA + 041Ch	DMA Channel 8 Interrupt Status Register	DMA8_INTSTA
DMA + 0420h	DMA Channel 8 Interrupt Acknowledge Register	DMA8_ACKINT
DMA + 0424h	DMA Channel 8 Remaining Length of Current Transfer	DMA8_RLCT
DMA + 0428h	DMA Channel 8 Bandwidth Limiter Register	DMA8_LIMITER
DMA + 0488h	DMA Channel 9 Wrap Point Address Register	DMA9_WPPT
DMA + 048Ch	DMA Channel 9 Wrap To Address Register	DMA9_WPTO
DMA + 0490h	DMA Channel 9 Transfer Count Register	DMA9_COUNT
DMA + 0494h	DMA Channel 9 Control Register	DMA9_CON
DMA + 0498h	DMA Channel 9 Start Register	DMA9_START
DMA + 049Ch	DMA Channel 9 Interrupt Status Register	DMA9_INTSTA
DMA + 04A0h	DMA Channel 9 Interrupt Acknowledge Register	DMA9_ACKINT
DMA + 04A4h	DMA Channel 9 Remaining Length of Current Transfer	DMA9_RLCT
DMA + 04A8h	DMA Channel 9 Bandwidth Limiter Register	DMA9_LIMITER
DMA + 04aCh	DMA Channel 9 Programmable Address Register	DMA9_PGMADDR
DMA + 0508h	DMA Channel 10 Wrap Point Address Register	DMA10_WPPT
DMA + 050Ch	DMA Channel 10 Wrap To Address Register	DMA10_WPTO
DMA + 0510h	DMA Channel 10 Transfer Count Register	DMA10_COUNT
DMA + 0514h	DMA Channel 10 Control Register	DMA10_CON
DMA + 0518h	DMA Channel 10 Start Register	DMA10_START
DMA + 051Ch	DMA Channel 10 Interrupt Status Register	DMA10_INTSTA
DMA + 0520h	DMA Channel 10 Interrupt Acknowledge Register	DMA10_ACKINT
DMA + 0524h	DMA Channel 10 Remaining Length of Current Transfer	DMA10_RLCT
DMA + 0528h	DMA Channel 10 Bandwidth Limiter Register	DMA10_LIMITER

© 2009 MediaTek Inc.

DMA + 052Ch	DMA Channel 10 Programmable Address Register	DMA10_PGMADDR
DMA + 0588h	DMA Channel 11 Wrap Point Address Register	DMA11_WPPT
DMA + 058Ch	DMA Channel 11 Wrap To Address Register	DMA11_WPTO
DMA + 0590h	DMA Channel 11 Transfer Count Register	DMA11_COUNT
DMA + 0594h	DMA Channel 11 Control Register	DMA11_CON
DMA + 0598h	DMA Channel 11 Start Register	DMA11_START
DMA + 059Ch	DMA Channel 11 Interrupt Status Register	DMA11_INTSTA
DMA + 05A0h	DMA Channel 11 Interrupt Acknowledge Register	DMA11_ACKINT
DMA + 05A4h	DMA Channel 11 Remaining Length of Current Transfer	DMA11_RLCT
DMA + 05A8h	DMA Channel 11 Bandwidth Limiter Register	DMA11_LIMITER
DMA + 05ACh	DMA Channel 11 Programmable Address Register	DMA11_PGMADDR
DMA + 0608h	DMA Channel 12 Wrap Point Address Register	DMA12_WPPT
DMA + 060Ch	DMA Channel 12 Wrap To Address Register	DMA12_WPTO
DMA + 0610h	DMA Channel 12 Transfer Count Register	DMA12_COUNT
DMA + 0614h	DMA Channel 12 Control Register	DMA12_CON
DMA + 0618h	DMA Channel 12 Start Register	DMA12_START
DMA + 061Ch	DMA Channel 12 Interrupt Status Register	DMA12_INTSTA
DMA + 0620h	DMA Channel 12 Interrupt Acknowledge Register	DMA12_ACKINT
DMA + 0624h	DMA Channel 12 Remaining Length of Current Transfer	DMA12_RLCT
DMA + 0628h	DMA Channel 12 Bandwidth Limiter Register	DMA12_LIMITER
DMA + 062Ch	DMA Channel 12 Programmable Address Register	DMA12_PGMADDR
DMA + 0688h	DMA Channel 13 Wrap Point Address Register	DMA13_WPPT
DMA + 068Ch	DMA Channel 13 Wrap To Address Register	DMA13_WPTO
DMA + 0690h	DMA Channel 13 Transfer Count Register	DMA13_COUNT
DMA + 0694h	DMA Channel 13 Control Register	DMA13_CON
DMA + 0698h	DMA Channel 13 Start Register	DMA13_START
DMA + 069Ch	DMA Channel 13 Interrupt Status Register	DMA13_INTSTA
DMA + 06A0h	DMA Channel 13 Interrupt Acknowledge Register	DMA13_ACKINT
DMA + 06A4h	DMA Channel 13 Remaining Length of Current Transfer	DMA13_RLCT
DMA + 06A8h	DMA Channel 13 Bandwidth Limiter Register	DMA13_LIMITER
DMA + 06ACh	DMA Channel 13 Programmable Address Register	DMA13_PGMADDR

© 2009 MediaTek Inc.

DMA + 0708h	DMA Channel 14 Wrap Point Address Register	DMA14_WPPT
DMA + 070Ch	DMA Channel 14 Wrap To Address Register	DMA14_WPTO
DMA + 0710h	DMA Channel 14 Transfer Count Register	DMA14_COUNT
DMA + 0714h	DMA Channel 14 Control Register	DMA14_CON
DMA + 0718h	DMA Channel 14 Start Register	DMA14_START
DMA + 071Ch	DMA Channel 14 Interrupt Status Register	DMA14_INTSTA
DMA + 0720h	DMA Channel 14 Interrupt Acknowledge Register	DMA14_ACKINT
DMA + 0724h	DMA Channel 14 Remaining Length of Current Transfer	DMA14_RLCT
DMA + 0728h	DMA Channel 14 Bandwidth Limiter Register	DMA14_LIMITER
DMA + 072Ch	DMA Channel 14 Programmable Address Register	DMA14_PGMADDR
DMA + 078Ch	DMA Channel 15 Wrap To Address Register	DMA15_WPTO
DMA + 0790h	DMA Channel 15 Transfer Count Register	DMA15_COUNT
DMA + 0794h	DMA Channel 15 Control Register	DMA15_CON
DMA + 0798h	DMA Channel 15 Start Register	DMA15_START
DMA + 079Ch	DMA Channel 15 Interrupt Status Register	DMA15_INTSTA
DMA + 07A0h	DMA Channel 15 Interrupt Acknowledge Register	DMA15_ACKINT
DMA + 07A4h	DMA Channel 15 Remaining Length of Current Transfer	DMA15_RLCT
DMA + 07A8h	DMA Channel 15 Bandwidth Limiter Register	DMA15_LIMITER
DMA + 07ACh	DMA Channel 15 Programmable Address Register	DMA15_PGMADDR
DMA + 0808h	DMA Channel 16 Wrap Point Address Register	DMA16_WPPT
DMA + 080Ch	DMA Channel 16 Wrap To Address Register	DMA16_WPTO
DMA + 0810h	DMA Channel 16 Transfer Count Register	DMA16_COUNT
DMA + 0814h	DMA Channel 16 Control Register	DMA16_CON
DMA + 0818h	DMA Channel 16 Start Register	DMA16_START
DMA + 081Ch	DMA Channel 16 Interrupt Status Register	DMA16_INTSTA
DMA + 0820h	DMA Channel 16 Interrupt Acknowledge Register	DMA16_ACKINT
DMA + 0824h	DMA Channel 16 Remaining Length of Current Transfer	DMA16_RLCT
DMA + 0828h	DMA Channel 16 Bandwidth Limiter Register	DMA16_LIMITER
DMA + 082Ch	DMA Channel 16 Programmable Address Register	DMA16_PGMADDR
DMA + 0890h	DMA Channel 17 Transfer Count Register	DMA17_COUNT
DMA + 0894h	DMA Channel 17 Control Register	DMA17_CON

DMA + 0898h	DMA Channel 17 Start Register	DMA17_START
DMA + 089Ch	DMA Channel 17 Interrupt Status Register	DMA17_INTSTA
DMA + 08A0h	DMA Channel 17 Interrupt Acknowledge Register	DMA17_ACKINT
DMA + 08A8h	DMA Channel 17 Bandwidth Limiter Register	DMA17_LIMITER
DMA + 08ACh	DMA Channel 17 Programmable Address Register	DMA17_PGMADDR
DMA + 08B0h	DMA Channel 17 Virtual FIFO Write Pointer	DMA17_WRPTR
DMA + 08B4h	DMA Channel 17 Virtual FIFO Read Pointer	DMA17_RDPTR
DMA + 08B8h	DMA Channel 17 Virtual FIFO Data Count	DMA17_FFCNT
DMA + 08BCh	DMA Channel 17 Virtual FIFO Status	DMA17_FFSTA
DMA + 08C0h	DMA Channel 17 Virtual FIFO Alert Length	DMA17_ALTLEN
DMA + 08C4h	DMA Channel 17 Virtual FIFO Size	DMA17_FFSIZE
DMA + 0910h	DMA Channel 18 Transfer Count Register	DMA18_COUNT
DMA + 0914h	DMA Channel 18 Control Register	DMA18_CON
DMA + 0918h	DMA Channel 18 Start Register	DMA18_START
DMA + 091Ch	DMA Channel 18 Interrupt Status Register	DMA18_INTSTA
DMA + 0920h	DMA Channel 18 Interrupt Acknowledge Register	DMA18_ACKINT
DMA + 0928h	DMA Channel 18 Bandwidth Limiter Register	DMA18_LIMITER
DMA + 092Ch	DMA Channel 18 Programmable Address Register	DMA18_PGMADDR
DMA + 0930h	DMA Channel 18 Virtual FIFO Write Pointer	DMA18_WRPTR
DMA + 0934h	DMA Channel 18 Virtual FIFO Read Pointer	DMA18_RDPTR
DMA + 0938h	DMA Channel 18 Virtual FIFO Data Count	DMA18_FFCNT
DMA + 093Ch	DMA Channel 18 Virtual FIFO Status	DMA18_FFSTA
DMA + 0940h	DMA Channel 18 Virtual FIFO Alert Length	DMA18_ALTLEN
DMA + 0944h	DMA Channel 18 Virtual FIFO Size	DMA18_FFSIZE
DMA + 0980h	DMA Channel 19 Transfer Count Register	DMA19_COUNT
DMA + 0984h	DMA Channel 19 Control Register	DMA19_CON
DMA + 0988h	DMA Channel 19 Start Register	DMA19_START
DMA + 098Ch	DMA Channel 19 Interrupt Status Register	DMA19_INTSTA
DMA + 09A0h	DMA Channel 19 Interrupt Acknowledge Register	DMA19_ACKINT
DMA + 09A8h	DMA Channel 19 Bandwidth Limiter Register	DMA19_LIMITER
DMA + 09ACh	DMA Channel 19 Programmable Address Register	DMA19_PGMADDR
DMA + 09B0h	DMA Channel 19 Virtual FIFO Write Pointer	DMA19_WRPTR
DMA + 09B4h	DMA Channel 19 Virtual FIFO Read Pointer	DMA19_RDPTR

© 2009 MediaTek Inc.

DMA + 09B8h	DMA Channel 19 Virtual FIFO Data Count	DMA19_FFCNT
DMA + 09BCh	DMA Channel 19 Virtual FIFO Status	DMA19_FFSTA
DMA + 09C0h	DMA Channel 19 Virtual FIFO Alert Length	DMA19_ALTLEN
DMA + 09C4h	DMA Channel 19 Virtual FIFO Size	DMA19_FFSIZE
DMA + 0A00h	DMA Channel 20 Transfer Count Register	DMA20_COUNT
DMA + 0A04h	DMA Channel 20 Control Register	DMA20_CON
DMA + 0A08h	DMA Channel 20 Start Register	DMA20_START
DMA + 0A0Ch	DMA Channel 20 Interrupt Status Register	DMA20_INTSTA
DMA + 0A20h	DMA Channel 20 Interrupt Acknowledge Register	DMA20_ACKINT
DMA + 0A28h	DMA Channel 20 Bandwidth Limiter Register	DMA20_LIMITER
DMA + 0A2Ch	DMA Channel 20 Programmable Address Register	DMA20_PGMADDR
DMA + 0A30h	DMA Channel 20 Virtual FIFO Write Pointer	DMA20_WRPTR
DMA + 0A34h	DMA Channel 20 Virtual FIFO Read Pointer	DMA20_RDPTR
DMA + 0A38h	DMA Channel 20 Virtual FIFO Data Count	DMA20_FFCNT
DMA + 0A3Ch	DMA Channel 20 Virtual FIFO Status	DMA20_FFSTA
DMA + 0A40h	DMA Channel 20 Virtual FIFO Alert Length	DMA20_ALTLEN
DMA + 0A44h	DMA Channel 20 Virtual FIFO Size	DMA20_FFSIZE
DMA + 0A90h	DMA Channel 21 Transfer Count Register	DMA21_COUNT
DMA + 0A94h	DMA Channel 21 Control Register	DMA21_CON
DMA + 0A98h	DMA Channel 21 Start Register	DMA21_START
DMA + 0A9Ch	DMA Channel 21 Interrupt Status Register	DMA21_INTSTA
DMA + 0AA0h	DMA Channel 21 Interrupt Acknowledge Register	DMA21_ACKINT
DMA + 0AA8h	DMA Channel 21 Bandwidth Limiter Register	DMA21_LIMITER
DMA + 0AACh	DMA Channel 21 Programmable Address Register	DMA21_PGMADDR
DMA + 0AB0h	DMA Channel 21 Virtual FIFO Write Pointer	DMA21_WRPTR
DMA + 0AB4h	DMA Channel 21 Virtual FIFO Read Pointer	DMA21_RDPTR
DMA + 0AB8h	DMA Channel 21 Virtual FIFO Data Count	DMA21_FFCNT
DMA + 0ABCh	DMA Channel 21 Virtual FIFO Status	DMA21_FFSTA
DMA + 0AC0h	DMA Channel 21 Virtual FIFO Alert Length	DMA21_ALTLEN
DMA + 0AC4h	DMA Channel 21 Virtual FIFO Size	DMA21_FFSIZE
DMA + 0B10h	DMA Channel 22 Transfer Count Register	DMA22_COUNT
DMA + 0B14h	DMA Channel 22 Control Register	DMA22_CON
DMA + 0B18h	DMA Channel 22 Start Register	DMA22_START
DMA + 0B1Ch	DMA Channel 22 Interrupt Status Register	DMA22_INTSTA

© 2009 MediaTek Inc.

DMA + 0B20h	DMA Channel 22 Interrupt Acknowledge Register	DMA22_ACKINT
DMA + 0B28h	DMA Channel 22 Bandwidth Limiter Register	DMA22_LIMITER
DMA + 0B2Ch	DMA Channel 22 Programmable Address Register	DMA22_PGMADDR
DMA + 0B30h	DMA Channel 22 Virtual FIFO Write Pointer	DMA22_WRPTR
DMA + 0B34h	DMA Channel 22 Virtual FIFO Read Pointer	DMA22_RDPTR
DMA + 0B38h	DMA Channel 22 Virtual FIFO Data Count	DMA22_FFCNT
DMA + 0B3Ch	DMA Channel 22 Virtual FIFO Status	DMA22_FFSTA
DMA + 0B40h	DMA Channel 22 Virtual FIFO Alert Length	DMA22_ALTLEN
DMA + 0B44h	DMA Channel 22 Virtual FIFO Size	DMA22_FFSIZE
DMA + 0B90h	DMA Channel 23 Transfer Count Register	DMA23_COUNT
DMA + 0B94h	DMA Channel 23 Control Register	DMA23_CON
DMA + 0B98h	DMA Channel 23 Start Register	DMA23_START
DMA + 0B9Ch	DMA Channel 23 Interrupt Status Register	DMA23_INTSTA
DMA + 0BA0h	DMA Channel 23 Interrupt Acknowledge Register	DMA23_ACKINT
DMA + 0BA8h	DMA Channel 23 Bandwidth Limiter Register	DMA23_LIMITER
DMA + 0BACh	DMA Channel 23 Programmable Address Register	DMA23_PGMADDR
DMA + 0BB0h	DMA Channel 23 Virtual FIFO Write Pointer	DMA23_WRPTR
DMA + 0BB4h	DMA Channel 23 Virtual FIFO Read Pointer	DMA23_RDPTR
DMA + 0BB8h	DMA Channel 23 Virtual FIFO Data Count	DMA23_FFCNT
DMA + 0BBCh	DMA Channel 23 Virtual FIFO Status	DMA23_FFSTA
DMA + 0BC0h	DMA Channel 23 Virtual FIFO Alert Length	DMA23_ALTLEN
DMA + 0BC4h	DMA Channel 23 Virtual FIFO Size	DMA23_FFSIZE
DMA + 0C10h	DMA Channel 24 Transfer Count Register	DMA24_COUNT
DMA + 0C14h	DMA Channel 24 Control Register	DMA24_CON
DMA + 0C18h	DMA Channel 24 Start Register	DMA24_START
DMA + 0C1Ch	DMA Channel 24 Interrupt Status Register	DMA24_INTSTA
DMA + 0C20h	DMA Channel 24 Interrupt Acknowledge Register	DMA24_ACKINT
DMA + 0C28h	DMA Channel 24 Bandwidth Limiter Register	DMA24_LIMITER
DMA + 0C2Ch	DMA Channel 24 Programmable Address Register	DMA24_PGMADDR
DMA + 0C30h	DMA Channel 24 Virtual FIFO Write Pointer	DMA24_WRPTR
DMA + 0C34h	DMA Channel 24 Virtual FIFO Read Pointer	DMA24_RDPTR
DMA + 0C38h	DMA Channel 24 Virtual FIFO Data Count	DMA24_FFCNT
DMA + 0C3Ch	DMA Channel 24 Virtual FIFO Status	DMA24_FFSTA

© 2009 MediaTek Inc.

DMA + 0C40h	DMA Channel 24 Virtual FIFO Alert Length	DMA24_ALTLEN
DMA + 0C44h	DMA Channel 24 Virtual FIFO Size	DMA24_FFSIZE

Table 8 DMA Controller Register Map

2.6.2 Register Definitions

Register programming tips:

- Start registers shall be cleared, when associated channels are being programmed.
- PGMADDR, i.e. programmable address, only exists in half-size DMA channels. If DIR in Control Register is high, PGMADDR represents Destination Address. Conversely, If DIR in Control Register is low, PGMADDR represents Source Address.
- Functions of ring-buffer and double-buffer memory data movement can be activated on either source side or destination side by programming DMA_WPPT & and DMA_WPTO, as well as setting WPEN in DMA_CON register high. WPSD in DMA_CON register determines the activated side.

DMA+0000h DMA Global Status Register DMA_GLBSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IT16	RUN16	IT15	RUN15	IT14	RUN14	IT13	RUN13	IT12	RUN12	IT11	RUN11	IT10	RUN10	IT9	RUN9
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IT8	RUN8	IT7	RUN7	IT6	RUN6	IT5	RUN5	IT4	RUN4	IT3	RUN3	IT2	RUN2	IT1	RUN1
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA+0004h DMA Global Status 2 Register DMA_GLBSTA2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IT24	RUN24	IT23	RUN23	IT22	RUN22	IT21	RUN21	IT20	RUN20	IT19	RUN19	IT18	RUN18	IT17	RUN17
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register helps software program keep track of the global status of DMA channels.

RUN_N DMA channel n status

0 Channel n is stopped or has completed the transfer already.

1 Channel n is currently running.

IT_N Interrupt status for channel n

0 No interrupt is generated.

1 An interrupt is pending and waiting for service.

© 2009 MediaTek Inc.

DMA+0028h DMA Global Bandwidth limiter Register DMA_GLBLIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name									GLBLIMITER							
Type									WO							
Reset									0							

Please refer to the expression in DMAn_LIMITER for detailed note. The value of DMA_GLBLIMITER is set to all DMA channels, from 1 to 15.

DMA+0080h DMA Channel 1 Source Address Register DMA1_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRC[31:16]															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SRC[15:0]															
Type	R/W															
Reset	0															

The above registers contain the base or current source address that the DMA channel is currently operating on. Writing to this register specifies the base address of transfer source for a DMA channel. Before programming these registers, the software program should make sure that STR in DMAn_START is set to 0; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value from which the DMA is reading. TCM is not accessible by DMA. The source addresses register for channel 1 to channel 8 are defined in registers listed in **Table 9**

SRC **SRC**[31:0] specifies the base or current address of transfer source for a DMA channel

WRITE Base address of transfer source

READ Address from which DMA is reading

Register Address	Register Function	Acronym
DMA + 0080h	DMA Channel 1 Source Address Register	DMA1_SRC
DMA + 0100h	DMA Channel 2 Source Address Register	DMA2_SRC
DMA + 0180h	DMA Channel 3 Source Address Register	DMA3_SRC
DMA + 0200h	DMA Channel 4 Source Address Register	DMA4_SRC
DMA + 0280h	DMA Channel 5 Source Address Register	DMA5_SRC
DMA + 0300h	DMA Channel 6 Source Address Register	DMA6_SRC
DMA + 0380h	DMA Channel 7 Source Address Register	DMA7_SRC
DMA + 0400h	DMA Channel 8 Source Address Register	DMA8_SRC

© 2009 MediaTek Inc.

Table 9 ***DMA Source Address Registers List***

DMA+0084hh DMA Channel 1 Destination Address Register DMA1_DST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DST[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DST[15:0]**															
Type	R/W															
Reset	0															

The above registers contain the base or current destination address that the DMA channel is currently operating on.. Writing to this register specifies the base address of the transfer destination for a DMA channel. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0'; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value to which the DMA is writing. TCM is not accessible by DMA. The destination addresses registers for channel 1 to channel 8 are defined in registers listed in **Table 10**

DST **DST**[31:0] specifies the base or current address of transfer destination for a DMA channel, i.e. channel 1~8

WRITE Base address of transfer destination.

READ Address to which DMA is writing.

Register Address	Register Function	Acronym
DMA + 0084h	DMA Channel 1 Destination Address Register	DMA1_DST
DMA + 0104h	DMA Channel 2 Destination Address Register	DMA2_DST
DMA + 0184h	DMA Channel 3 Destination Address Register	DMA3_DST
DMA + 0204h	DMA Channel 4 Destination Address Register	DMA4_DST
DMA + 0284h	DMA Channel 5 Destination Address Register	DMA5_DST
DMA + 0304h	DMA Channel 6 Destination Address Register	DMA6_DST
DMA + 0384h	DMA Channel 7 Destination Address Register	DMA7_DST
DMA + 0404h	DMA Channel 8 Destination Address Register	DMA8_DST

Table 10 ***DMA Destination Address Registers List***

DMA+0088h DMA Channel 1 Wrap Point Count Register DMA1_WPPT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WPPT[15:0]**															
Type	R/W															

© 2009 MediaTek Inc.

Reset	0

The above registers are to specify the transfer count required to perform before the jump point. This can be used to support ring buffer or double buffer style memory accesses. To enable this function, two control bits, WPEN and WPSD, in DMA control register must be programmed. See the following register description for more details. If the transfercounter in the DMA engine matches this value, an address jump occurs, and the next address is the address specified in DMAn_WPTO. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON is set. Note that the total size of data specify in the wrap point count in a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. WPPT x SIZE. The wrap point addresses registers for channel 1 to channel 16 are defined in registers listed in **Table 11**

WPPT **WPPT**[15:0] specifies the amount of the transfer count from start to jumping point for a DMA channel, i.e. channel 1 – 16.

WRITE Wrap point transfer count.

READ Value set by the programmer.

Register Address	Register Function	Acronym
DMA + 0088h	DMA Channel 1 Wrap Point Address Register	DMA1_WPPT
DMA + 0108h	DMA Channel 2 Wrap Point Address Register	DMA2_WPPT
DMA + 0188h	DMA Channel 3 Wrap Point Address Register	DMA3_WPPT
DMA + 0208h	DMA Channel 4 Wrap Point Address Register	DMA4_WPPT
DMA + 0288h	DMA Channel 5 Wrap Point Address Register	DMA5_WPPT
DMA + 0308h	DMA Channel 6 Wrap Point Address Register	DMA6_WPPT
DMA + 0388h	DMA Channel 7 Wrap Point Address Register	DMA7_WPPT
DMA + 0408h	DMA Channel 8 Wrap Point Address Register	DMA8_WPPT
DMA + 0488h	DMA Channel 9 Wrap Point Address Register	DMA9_WPPT
DMA + 0508h	DMA Channel 10 Wrap Point Address Register	DMA10_WPPT
DMA + 0588h	DMA Channel 11 Wrap Point Address Register	DMA11_WPPT
DMA + 0608h	DMA Channel 12 Wrap Point Address Register	DMA12_WPPT
DMA + 0688h	DMA Channel 13 Wrap Point Address Register	DMA13_WPPT
DMA + 0708h	DMA Channel 14 Wrap Point Address Register	DMA14_WPPT
DMA + 0788h	DMA Channel 15 Wrap Point Address Register	DMA15_WPPT
DMA + 0808h	DMA Channel 16 Wrap Point Address Register	DMA16_WPPT

Table 11 ***DMA Wrap Point Address Registers List***

DMA+0080C0h DMA Channel 1 Wrap To Address Register DMA1_WPTO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WPTO[31:16]**															
Type	R/W															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WPTO[15:0]**															
Type	R/W															
Reset	0															

The above registers specify the address of the jump destination of a given DMA transfer to support ring buffer or double buffer style memory accesses. To enable this function, set the two control bits, WPEN and WPSD, in the DMA control register . See the following register description for more details. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON should be set. The wrap to addresses registers for channel 1 to channel 16 are defined in registers listed in **Table 12**

WPTO **WPTO**[31:0] specifies the address of the jump point for a DMA channel, i.e. channel 1 – 16.
WRITE Address of the jump destination.
READ Value set by the programmer.

Register Address	Register Function	Acronym
DMA + 008Ch	DMA Channel 1 Wrap To Address Register	DMA1_WPTO
DMA + 010Ch	DMA Channel 2 Wrap To Address Register	DMA2_WPTO
DMA + 018Ch	DMA Channel 3 Wrap To Address Register	DMA3_WPTO
DMA + 020Ch	DMA Channel 4 Wrap To Address Register	DMA4_WPTO
DMA + 028Ch	DMA Channel 5 Wrap To Address Register	DMA5_WPTO
DMA + 030Ch	DMA Channel 6 Wrap To Address Register	DMA6_WPTO
DMA + 038Ch	DMA Channel 7 Wrap To Address Register	DMA7_WPTO
DMA + 040Ch	DMA Channel 8 Wrap To Address Register	DMA8_WPTO
DMA + 048Ch	DMA Channel 9 Wrap To Address Register	DMA9_WPTO
DMA + 050Ch	DMA Channel 10 Wrap To Address Register	DMA10_WPTO
DMA + 058Ch	DMA Channel 11 Wrap To Address Register	DMA11_WPTO
DMA + 060Ch	DMA Channel 12 Wrap To Address Register	DMA12_WPTO
DMA + 068Ch	DMA Channel 13 Wrap To Address Register	DMA13_WPTO
DMA + 070Ch	DMA Channel 14 Wrap To Address Register	DMA14_WPTO
DMA + 078Ch	DMA Channel 15 Wrap To Address Register	DMA15_WPTO
DMA + 080Ch	DMA Channel 16 Wrap To Address Register	DMA16_WPTO

Table 12 ***DMA Wrap To Address Registers List***

DMA+00810h DMA Channel 1 Transfer Count Register DMA1_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LEN**															
Type	R/W															
Reset	0															

This register specifies the amount of total transfer count that the DMA channel is required to perform. Upon completion, the DMA channel generates an interrupt request to the processor while ITEN in DMAn_CON is set as ‘1’. Note that the total size of data being transferred by a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. LEN x SIZE.

For virtual FIFO DMA, this register is used to configure the RX threshold and TX threshold. Interrupt is triggered while FIFO count >= RX threshold in RX path or FIFO count =< TX threshold in TX path. Note that ITEN bit in DMA_CON register shall be set, or no interrupt is issued. The transfer count registers for channel 1 to channel 24 are defined in registers listed in **Table 13**

LEN The amount of total transfer count

Register Address	Register Function	Acronym
DMA + 0090h	DMA Channel 1 Transfer Count Address Register	DMA1_COUNT
DMA + 0110h	DMA Channel 2 Transfer Count Address Register	DMA2_COUNT
DMA + 0190h	DMA Channel 3 Transfer Count Address Register	DMA3_COUNT
DMA + 0210h	DMA Channel 4 Transfer Count Address Register	DMA4_COUNT
DMA + 0290h	DMA Channel 5 Transfer Count Address Register	DMA5_COUNT
DMA + 0310h	DMA Channel 6 Transfer Count Address Register	DMA6_COUNT
DMA + 0390h	DMA Channel 7 Transfer Count Address Register	DMA7_COUNT
DMA + 0410h	DMA Channel 8 Transfer Count Address Register	DMA8_COUNT
DMA + 0490h	DMA Channel 9 Transfer Count Address Register	DMA9_COUNT
DMA + 0510h	DMA Channel 10 Transfer Count Address Register	DMA10_COUNT
DMA + 0590h	DMA Channel 11 Transfer Count Address Register	DMA11_COUNT
DMA + 0610h	DMA Channel 12 Transfer Count Address Register	DMA12_COUNT
DMA + 0690h	DMA Channel 13 Transfer Count Address Register	DMA13_COUNT
DMA + 0710h	DMA Channel 14 Transfer Count Address Register	DMA14_COUNT
DMA + 0790h	DMA Channel 15 Transfer Count Address Register	DMA15_COUNT
DMA + 0810h	DMA Channel 16 Transfer Count Address Register	DMA16_COUNT
DMA + 0890h	DMA Channel 17 Transfer Count Address Register	DMA17_COUNT
DMA + 0910h	DMA Channel 18Transfer Count Address Register	DMA18_COUNT
DMA + 0990h	DMA Channel 19 Transfer Count Address Register	DMA19_COUNT
DMA + 0A10h	DMA Channel 20 Transfer Count Address Register	DMA20_COUNT
DMA + 0A90h	DMA Channel 21 Transfer Count Address Register	DMA21_COUNT
DMA + 0B10h	DMA Channel 22 Transfer Count Address Register	DMA22_COUNT
DMA + 0B90h	DMA Channel 23 Transfer Count Address Register	DMA23_COUNT

 © 2009 MediaTek Inc.

DMA + 0C10h	DMA Channel 24 Transfer Count Address Register	DMA24_COUNT

Table 13 ***DMA Transfer Count Registers List***

DMA+00814h DMA Channel 1 Control Register DMA1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								MAS						DIR	WPEN	WPSD
Type								R/W						R/W	R/W	R/W
Reset								0						0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ITEN					BURST					B2W	DRQ	DINC	SINC	SIZE	
Type	R/W					R/W					R/W	R/W	R/W	R/W	R/W	
Reset	0					0					0	0	0	0	0	

This register contains all the available control schemes for a DMA channel that is ready for software programmer to configure. Note that all these fields cannot be changed while DMA transfer is in progress or an unexpected situation may occur. The transfer count registers for channel 1 to channel 24 are defined in registers listed in **Table 14**.

SIZE Data size within the confine of a bus cycle per transfer.
These bits confines the data transfer size between source and destination to the specified value for individual bus cycle. The size is in terms of byte and has maximum value of 4 bytes. It is mainly decided by the data width of a DMA master.

- **00** Byte transfer/1 byte
- **01** Half-word transfer/2 bytes
- **10** Word transfer/4 bytes
- **11** Reserved

SINC Incremental source address. Source addresses increase every transfer. If the setting of SIZE is Byte, Source addresses increase by 1 every single transfer. If Half-Word, increase by 2; and if Word, increase by 4.

- **0** Disable
- **1** Enable

DINC Incremental destination address. Destination addresses increase every transfer. If the setting of SIZE is Byte, Destination addresses increase by 1 every single transfer. If Half-Word, increase by 2; and Iif Word, increase by 4.

- **0** Disable
- **1** Enable

DREQ Throttle and handshake control for DMA transfer

- **0** No throttle control during DMA transfer or transfers occurred only between memories
- 1 Hardware handshake management
  The DMA master is able to throttle down the transfer rate by way of request-grant handshake.

B2W Word to Byte or Byte to Word transfer for the applications of transferring non-word-aligned-address data to word-aligned-address data. Note that BURST is set to 4-beat burst while enabling this function and the SIZE is set to Byte.
NO effect on channel 1 – 8 & 17 - 24.

0 Disable
1 Enable

BURST Transfer Type. Burst-type transfers have better bus efficiency. Mass data movement is recommended to use this kind of transfer. However, note that burst-type transfer does not stop until all of the beats in a burst are completed or transfer length is reached. FIFO threshold of peripherals must be configured carefully while being used to move data from/to the peripherals.
What transfer type can be used is restricted by the SIZE. If SIZE is 00b, i.e. byte transfer, all of the four transfer types can be used. If SIZE is 01b, i.e. half-word transfer, 16-beat incrementing burst cannot be used. If SIZE is 10b, i.e. word transfer, only single and 4-beat incrementing burst can be used.
NO effect on channel 17 - 24.
000 Single
001 Reserved
010 4-beat incrementing burst
011 Reserved
100 8-beat incrementing burst
101 Reserved
110 16-beat incrementing burst
111 Reserved

ITEN DMA transfer completion interrupt enable.
0 Disable
1 Enable

WPSD The side using address-wrapping function. Only one side of a DMA channel can activate address-wrapping function at a time.
NO effect on channel 17-24
0 Address-wrapping on source.
1 Address-wrapping on destination.

WPEN Address-wrapping for ring buffer and double buffer. The next address of DMA jumps to WRAP TO address when the current address matches WRAP POINT count.
NO effect on channel 17-24
0 Disable
1 Enable

DIR Directions of DMA transfer for half-size and Virtual FIFO DMA channels, i.e. channels 4~14. The direction is from the perspective of the DMA masters. WRITE means read from master and then write to the address specified in DMA_PGMADDR, and vice versa.
NO effect on channel 1-8
0 Read
1 Write

MAS Master selection. Specifies which master occupies this DMA channel. Once assigned to certain master, the corresponding DREQ and DACK are connected. For half-size and Virtual FIFO DMA channels, i.e. channels 9 ~ 15, a predefined address is assigned as well.
00000 SIM2

00001 MSDC1
00010 MSDC2
00011 IrDA TX
00100 IrDA RX
00101 UART1 TX
00110 UART1 RX
00111 UART2 TX
01000 UART2 RX
01001 UART3 TX
01010 UART3 RX
01011 NFI TX
01100 NFI RX
01101 VFE
01110 I2C TX
01111 I2C RX
10000 UART4 TX
10001 UART4 RX
10010 MSDC3
OTHERS Reserved

Register Address	Register Function	Acronym
DMA + 0094h	DMA Channel 1 Control Register	DMA1_CON
DMA + 0114h	DMA Channel 2 Control Register	DMA2_CON
DMA + 0194h	DMA Channel 3 Control Register	DMA3_CON
DMA + 0214h	DMA Channel 4 Control Register	DMA4_CON
DMA + 0294h	DMA Channel 5 Control Register	DMA5_CON
DMA + 0314h	DMA Channel 6 Control Register	DMA6_CON
DMA + 0394h	DMA Channel 7 Control Register	DMA7_CON
DMA + 0414h	DMA Channel 8 Control Register	DMA8_CON
DMA + 0494h	DMA Channel 9 Control Register	DMA9_CON
DMA + 0514h	DMA Channel 10 Control Register	DMA10_CON
DMA + 0594h	DMA Channel 11 Control Register	DMA11_CON
DMA + 0614h	DMA Channel 12 Control Register	DMA12_CON
DMA + 0694h	DMA Channel 13 Control Register	DMA13_CON
DMA + 0714h	DMA Channel 14 Control Register	DMA14_CON
DMA + 0794h	DMA Channel 15 Control Register	DMA15_CON
DMA + 0814h	DMA Channel 16 Control Register	DMA16_CON
DMA + 0894h	DMA Channel 17 Control Register	DMA17_CON
DMA + 0914h	DMA Channel 18Control Register	DMA18_CON

© 2009 MediaTek Inc.

DMA + 0994h	DMA Channel 19 Control Register	DMA19_CON
DMA + 0A14h	DMA Channel 20 Control Register	DMA20_CON
DMA + 0A94h	DMA Channel 21 Control Register	DMA21_CON
DMA + 0B14h	DMA Channel 22 Control Register	DMA22_CON
DMA + 0B94h	DMA Channel 23 Control Register	DMA23_CON
DMA + 0C14h	DMA Channel 24 Control Register	DMA24_CON

Table 14 ***DMA Control Registers List***

DMA+0098h DMA Channel 1 Start Register DMA1_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**STR**															
Type	R/W															
Reset	0															

This register controls the activity of a DMA channel. When **STR** is changed from 0 to 1, the DMA channel starts to work. Note that prior to setting STR to “1”, all the configurations should be done by giving proper value to the registers. Note also that once the STR is set to “1”, the hardware does not clear it automatically no matter if the DMA channel accomplishes the DMA transfer or not. In other words, the value of **STR** stays “1” regardless of the completion of DMA transfer. Therefore, the software program should be sure to clear **STR** to “0” for starting another transfer for the same DMA channel. If this bit is cleared to “0” during DMA transfer is active, software should polling DMA_GLBSTA **RUN_N** after this bit is cleared to ensure current DMA transfer is terminated by DMA engine. The DMA Start registers for channel 1 to channel 24 are defined in registers listed in **Table 15**

STR Start control for a DMA channel.

- **0** The DMA channel is stopped.
- **1** The DMA channel is started and running

Register Address	**Register Function**	**Acronym**
DMA + 0098h	DMA Channel 1 Start Register	DMA1_START
DMA + 0118h	DMA Channel 2 Start Register	DMA2_START
DMA + 0198h	DMA Channel 3 Start Register	DMA3_START
DMA + 0218h	DMA Channel 4 Start Register	DMA4_START
DMA + 0298h	DMA Channel 5 Start Register	DMA5_START
DMA + 0318h	DMA Channel 6 Start Register	DMA6_START
DMA + 0398h	DMA Channel 7 Start Register	DMA7_START
DMA + 0418h	DMA Channel 8 Start Register	DMA8_START
DMA + 0498h	DMA Channel 9 Start Register	DMA9_START

DMA + 0518h	DMA Channel 10 Start Register	DMA10_START
DMA + 0598h	DMA Channel 11 Start Register	DMA11_START
DMA + 0618h	DMA Channel 12 Start Register	DMA12_START
DMA + 0698h	DMA Channel 13 Start Register	DMA13_START
DMA + 0718h	DMA Channel 14 Start Register	DMA14_START
DMA + 0798h	DMA Channel 15 Start Register	DMA15_START
DMA + 0818h	DMA Channel 16 Start Register	DMA16_START
DMA + 0898h	DMA Channel 17 Start Register	DMA17_START
DMA + 0918h	DMA Channel 18Start Register	DMA18_START
DMA + 0998h	DMA Channel 19 Start Register	DMA19_START
DMA + 0A18h	DMA Channel 20 Start Register	DMA20_START
DMA + 0A98h	DMA Channel 21 Start Register	DMA21_START
DMA + 0B18h	DMA Channel 22 Start Register	DMA22_START
DMA + 0B98h	DMA Channel 23 Start Register	DMA23_START
DMA + 0C18h	DMA Channel 24 Start Register	DMA24_START

Table 15 ***DMA Start Registers List***

.

DMA+009Ch DMA Channel 1 Interrupt Status Register DMA1_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT															
Type	RO															
Reset	0															

This register shows the interrupt status of a DMA channel. It has the same value as DMA_GLBSTA. The DMA Interrupt Status registers for channel 1 to channel 24 are defined in registers listed in **Table 16**

INT Interrupt Status for DMA Channel

- **0** No interrupt request is generated.
- **1** One interrupt request is pending and waiting for service.

Register Address	Register Function	Acronym
DMA + 009Ch	DMA Channel 1 Interrupt Status Register	DMA1_INTSTA
DMA + 011Ch	DMA Channel 2 Interrupt Status Register	DMA2_INTSTA
DMA + 019Ch	DMA Channel 3 Interrupt Status Register	DMA3_INTSTA
DMA + 021Ch	DMA Channel 4 Interrupt Status Register	DMA4_INTSTA
DMA + 029Ch	DMA Channel 5 Interrupt Status Register	DMA5_INTSTA

DMA + 031Ch	DMA Channel 6 Interrupt Status Register	DMA6_INTSTA
DMA + 039Ch	DMA Channel 7 Interrupt Status Register	DMA7_INTSTA
DMA + 041Ch	DMA Channel 8 Interrupt Status Register	DMA8_INTSTA
DMA + 049Ch	DMA Channel 9 Interrupt Status Register	DMA9_INTSTA
DMA + 051Ch	DMA Channel 10 Interrupt Status Register	DMA10_INTSTA
DMA + 059Ch	DMA Channel 11 Interrupt Status Register	DMA11_INTSTA
DMA + 061Ch	DMA Channel 12 Interrupt Status Register	DMA12_INTSTA
DMA + 069Ch	DMA Channel 13 Interrupt Status Register	DMA13_INTSTA
DMA + 071Ch	DMA Channel 14 Interrupt Status Register	DMA14_INTSTA
DMA + 079Ch	DMA Channel 15 Interrupt Status Register	DMA15_INTSTA
DMA + 081Ch	DMA Channel 16 Interrupt Status Register	DMA16_INTSTA
DMA + 089Ch	DMA Channel 17 Interrupt Status Register	DMA17_INTSTA
DMA + 091Ch	DMA Channel 18Interrupt Status Register	DMA18_INTSTA
DMA + 099Ch	DMA Channel 19 Interrupt Status Register	DMA19_INTSTA
DMA + 0A1Ch	DMA Channel 20 Interrupt Status Register	DMA20_INTSTA
DMA + 0A9Ch	DMA Channel 21 Interrupt Status Register	DMA21_INTSTA
DMA + 0B1Ch	DMA Channel 22 Interrupt Status Register	DMA22_INTSTA
DMA + 0B9Ch	DMA Channel 23 Interrupt Status Register	DMA23_INTSTA
DMA + 0C1Ch	DMA Channel 24 Interrupt Status Register	DMA24_INTSTA

Table 16 ***DMA Interrupt Status Registers List***

DMA+00A0h DMA Channel n Interrupt Acknowledge Register DMA1_ACKINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ACK															
Type	WO															
Reset	0															

This register is used to acknowledge the current interrupt request associated with the completion event of a DMA channel by software program. Note that this is a write-only register, and any read to it returns a value of "0". The DMA Interrupt Acknowledge registers for channel 1 to channel 24 are defined in registers listed in **Table 17**

ACK Interrupt acknowledge for the DMA channel

- **0** No effect
- **1** Interrupt request is acknowledged and should be relinquished.

Register Address	Register Function	Acronym

DMA + 00A0h	DMA Channel 1 Interrupt Acknowledge Register	DMA1_ACKINT
DMA + 0120h	DMA Channel 2 Interrupt Acknowledge Register	DMA2_ACKINT
DMA + 01A0h	DMA Channel 3 Interrupt Acknowledge Register	DMA3_ACKINT
DMA + 0220h	DMA Channel 4 Interrupt Acknowledge Register	DMA4_ACKINT
DMA + 02A0h	DMA Channel 5 Interrupt Acknowledge Register	DMA5_ACKINT
DMA + 0320h	DMA Channel 6 Interrupt Acknowledge Register	DMA6_ACKINT
DMA + 03A0h	DMA Channel 7 Interrupt Acknowledge Register	DMA7_ACKINT
DMA + 0420h	DMA Channel 8 Interrupt Acknowledge Register	DMA8_ACKINT
DMA + 04A0h	DMA Channel 9 Interrupt Acknowledge Register	DMA9_ACKINT
DMA + 0520h	DMA Channel 10 Interrupt Acknowledge Register	DMA10_ACKINT
DMA + 05A0h	DMA Channel 11 Interrupt Acknowledge Register	DMA11_ACKINT
DMA + 0620h	DMA Channel 12 Interrupt Acknowledge Register	DMA12_ACKINT
DMA + 06A0h	DMA Channel 13 Interrupt Acknowledge Register	DMA13_ACKINT
DMA + 0720h	DMA Channel 14 Interrupt Acknowledge Register	DMA14_ACKINT
DMA + 07A0h	DMA Channel 15 Interrupt Acknowledge Register	DMA15_ACKINT
DMA + 0820h	DMA Channel 16 Interrupt Acknowledge Register	DMA16_ACKINT
DMA + 08A0h	DMA Channel 17 Interrupt Acknowledge Register	DMA17_ACKINT
DMA + 0920h	DMA Channel 18Interrupt Acknowledge Register	DMA18_ACKINT
DMA + 09A0h	DMA Channel 19 Interrupt Acknowledge Register	DMA19_ACKINT
DMA + 0A20h	DMA Channel 20 Interrupt Acknowledge Register	DMA20_ACKINT
DMA + 0AA0h	DMA Channel 21 Interrupt Acknowledge Register	DMA21_ACKINT
DMA + 0B20h	DMA Channel 22 Interrupt Acknowledge Register	DMA22_ACKINT
DMA + 0BA0h	DMA Channel 23 Interrupt Acknowledge Register	DMA23_ACKINT
DMA + 0C20h	DMA Channel 24 Interrupt Acknowledge Register	DMA24_ACKINT

Table 17 ***DMA Interrupt Acknowledge Registers List***

DMA+00A4h DMA Channel 1 Remaining Length of Current Transfer DMA1_RLCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**RLCT**															
Type	RO															
Reset	0															

This register is to reflect the left count of the transfer. Note that this value is transfer count not the transfer data size. The DMA Remaining Length of Current Transfer registers for channel 1 to channel 24 are defined in registers listed in **Table 18**

Register Address	Register Function	Acronym
DMA + 00A4h	DMA Channel 1 Remaining Length of Current Transfer Register	DMA1_RLCT
DMA + 0124h	DMA Channel 2 Remaining Length of Current Transfer Register	DMA2_RLCT
DMA + 01A4h	DMA Channel 3 Remaining Length of Current Transfer Register	DMA3_RLCT
DMA + 0224h	DMA Channel 4 Remaining Length of Current Transfer Register	DMA4_RLCT
DMA + 02A4h	DMA Channel 5 Remaining Length of Current Transfer Register	DMA5_RLCT
DMA + 0324h	DMA Channel 6 Remaining Length of Current Transfer Register	DMA6_RLCT
DMA + 03A4h	DMA Channel 7 Remaining Length of Current Transfer Register	DMA7_RLCT
DMA + 0424h	DMA Channel 8 Remaining Length of Current Transfer Register	DMA8_RLCT
DMA + 04A4h	DMA Channel 9 Remaining Length of Current Transfer Register	DMA9_RLCT
DMA + 0524h	DMA Channel 10 Remaining Length of Current Transfer Register	DMA10_RLCT
DMA + 05A4h	DMA Channel 11 Remaining Length of Current Transfer Register	DMA11_RLCT
DMA + 0624h	DMA Channel 12 Remaining Length of Current Transfer Register	DMA12_RLCT
DMA + 06A4h	DMA Channel 13 Remaining Length of Current Transfer Register	DMA13_RLCT
DMA + 0724h	DMA Channel 14 Remaining Length of Current Transfer Register	DMA14_RLCT
DMA + 07A4h	DMA Channel 15 Remaining Length of Current Transfer Register	DMA15_RLCT
DMA + 0824h	DMA Channel 16 Remaining Length of Current Transfer Register	DMA16_RLCT
DMA + 08A4h	DMA Channel 17 Remaining Length of Current Transfer Register	DMA17_RLCT
DMA + 0924h	DMA Channel 18Remaining Length of Current Transfer Register	DMA18_RLCT
DMA + 09A4h	DMA Channel 19 Remaining Length of Current Transfer Register	DMA19_RLCT
DMA + 0A24h	DMA Channel 20 Remaining Length of Current Transfer Register	DMA20_RLCT
DMA + 0AA4h	DMA Channel 21 Remaining Length of Current Transfer	DMA21_RLCT

	Register	
DMA + 0B24h	DMA Channel 22 Remaining Length of Current Transfer Register	DMA22_RLCT
DMA + 0BA4h	DMA Channel 23 Remaining Length of Current Transfer Register	DMA23_RLCT
DMA + 0C24h	DMA Channel 24 Remaining Length of Current Transfer Register	DMA24_RLCT

Table 18 ***DMA Remaining Length of Current Transfer Registers List***

DMA+00A8h DMA Channel 1 Bandwidth limiter Register DMA1_LIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name									**LIMITER**							
Type									R/W							
Reset									0							

This register is to suppress the Bus utilization of the DMA channel. The value is from 0 to 255. 0 means no limitation, and 255 means totally banned. The value between 0 and 255 means certain DMA can have permission to use AHB every (4 X n) AHB clock cycles.

Note that it is not recommended to limit the Bus utilization of the DMA channels because this increases the latency of response to the masters, and the transfer rate decreases as well. Before using it, programmer must make sure that the bus masters have some protective mechanism to avoid entering the wrong states. The DMA Bandwidth Limiter registers for channel 1 to channel 24 are defined in registers listed in Table 19

LIMITER from 0 to 255. 0 means no limitation, 255 means totally banned, and others mean Bus access permission every (4 X n) AHB clock.

Register Address	Register Function	Acronym
DMA + 00A8h	DMA Channel 1 Bandwidth Limiter Register	DMA1_LIMITER
DMA + 0128h	DMA Channel 2 Bandwidth Limiter Register	DMA2_LIMITER
DMA + 01A8h	DMA Channel 3 Bandwidth Limiter Register	DMA3_LIMITER
DMA + 0228h	DMA Channel 4 Bandwidth Limiter Register	DMA4_LIMITER
DMA + 02A8h	DMA Channel 5 Bandwidth Limiter Register	DMA5_LIMITER
DMA + 0328h	DMA Channel 6 Bandwidth Limiter Register	DMA6_LIMITER
DMA + 03A8h	DMA Channel 7 Bandwidth Limiter Register	DMA7_LIMITER
DMA + 0428h	DMA Channel 8 Bandwidth Limiter Register	DMA8_LIMITER
DMA + 04A8h	DMA Channel 9 Bandwidth Limiter Register	DMA9_LIMITER
DMA + 0528h	DMA Channel 10 Bandwidth Limiter Register	DMA10_LIMITER
DMA + 05A8h	DMA Channel 11 Bandwidth Limiter Register	DMA11_LIMITER
DMA + 0628h	DMA Channel 12 Bandwidth Limiter Register	DMA12_LIMITER

© 2009 MediaTek Inc.

DMA + 06A8h	DMA Channel 13 Bandwidth Limiter Register	DMA13_LIMITER
DMA + 0728h	DMA Channel 14 Bandwidth Limiter Register	DMA14_LIMITER
DMA + 07A8h	DMA Channel 15 Bandwidth Limiter Register	DMA15_LIMITER
DMA + 0828h	DMA Channel 16 Bandwidth Limiter Register	DMA16_LIMITER
DMA + 08A8h	DMA Channel 17 Bandwidth Limiter Register	DMA17_LIMITER
DMA + 0928h	DMA Channel 18Bandwidth Limiter Register	DMA18_LIMITER
DMA + 09A8h	DMA Channel 19 Bandwidth Limiter Register	DMA19_LIMITER
DMA + 0A28h	DMA Channel 20 Bandwidth Limiter Register	DMA20_LIMITER
DMA + 0AA8h	DMA Channel 21 Bandwidth Limiter Register	DMA21_LIMITER
DMA + 0B28h	DMA Channel 22 Bandwidth Limiter Register	DMA22_LIMITER
DMA + 0BA8h	DMA Channel 23 Bandwidth Limiter Register	DMA23_LIMITER
DMA + 0C28h	DMA Channel 24 Bandwidth Limiter Register	DMA24_LIMITER

Table 19 DMA Bandwidth Limiter Registers List

DMA+04ACh DMA Channel 9 Programmable Address Register DMA9_PGMADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PGMADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PGMADDR[15:0]**															
Type	R/W															
Reset	0															

The above registers specify the address for a half-size DMA channel. This address represents a source address if DIR in DMA_CON is set to 0, and represents a destination address if DIR in DMA_CON is set to 1. Before being able to program these register, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. The DMA Programmable Address registers for channel 1 to channel 24 are defined in registers listed in Table 20

Note that n is from 9 to 24 and PGMADDR can't be TCM address. TCM is not accessible by DMA.

PGMADDR **PGMADDR**[31:0] specifies the addresses for a half-size or a Virtual FIFO DMA channel, i.e. channel 9 – 24.

WRITE Base address of transfer source or destination according to DIR bit.

READ Current address of the transfer.

Register Address	Register Function	Acronym
DMA + 04ACh	DMA Channel 9 Programmable Address Register	DMA9_PGMADDR
DMA + 052Ch	DMA Channel 10 Programmable Address Register	DMA10_PGMADDR
DMA + 05ACh	DMA Channel 11 Programmable Address Register	DMA11_PGMADDR

DMA + 062Ch	DMA Channel 12 Programmable Address Register	DMA12_PGMADDR
DMA + 06ACh	DMA Channel 13 Programmable Address Register	DMA13_PGMADDR
DMA + 072Ch	DMA Channel 14 Programmable Address Register	DMA14_PGMADDR
DMA + 07ACh	DMA Channel 15 Programmable Address Register	DMA15_PGMADDR
DMA + 082Ch	DMA Channel 16 Programmable Address Register	DMA16_PGMADDR
DMA + 08ACh	DMA Channel 17 Programmable Address Register	DMA17_PGMADDR
DMA + 092Ch	DMA Channel 18 Programmable Address Register	DMA18_PGMADDR
DMA + 09ACh	DMA Channel 19 Programmable Address Register	DMA19_PGMADDR
DMA + 0A2Ch	DMA Channel 20 Programmable Address Register	DMA20_PGMADDR
DMA + 0AACh	DMA Channel 21 Programmable Address Register	DMA21_PGMADDR
DMA + 0B2Ch	DMA Channel 22 Programmable Address Register	DMA22_PGMADDR
DMA + 0BACh	DMA Channel 23 Programmable Address Register	DMA23_PGMADDR
DMA + 0C2Ch	DMA Channel 24 Programmable Address Register	DMA24_PGMADDR

Table 205 DMA Programmable Address Registers List

DMA+08B0h DMA Channel 17 Virtual FIFO Write Pointer Register DMA17_WRPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WRPTR[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WRPTR[15:0]**															
Type	RO															

The DMA Virtual FIFO Write Pointer registers for channel 17 to channel 24 are defined in registers listed in Table 21

WRPTR Virtual FIFO Write Pointer.

Register Address	Register Function	Acronym
DMA + 08B0h	DMA Channel 17 Virtual FIFO Write Pointer Register	DMA17_WRPTR
DMA + 0930h	DMA Channel 18 Virtual FIFO Write Pointer Register	DMA18_WRPTR
DMA + 09B0h	DMA Channel 19 Virtual FIFO Write Pointer Register	DMA19_WRPTR
DMA + 0A30h	DMA Channel 20 Virtual FIFO Write Pointer Register	DMA20_WRPTR
DMA + 0AB0h	DMA Channel 21 Virtual FIFO Write Pointer Register	DMA21_WRPTR
DMA + 0B30h	DMA Channel 22 Virtual FIFO Write Pointer Register	DMA22_WRPTR
DMA + 0BB0h	DMA Channel 23 Virtual FIFO Write Pointer Register	DMA23_WRPTR
DMA + 0C30h	DMA Channel 24 Virtual FIFO Write Pointer Register	DMA24_WRPTR

Table 21 DMA Virtual FIFO Write Pointer Registers List

DMA+08B4h DMA Channel 17 Virtual FIFO Read Pointer Register DMA17_RDPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RDPTR[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RDPTR[15:0]															
Type	RO															

The DMA Virtual FIFO Read Pointer registers for channel 17 to channel 24 are defined in registers listed in Table 22

RDPTR Virtual FIFO Read Pointer.

Register Address	Register Function	Acronym
DMA + 08B4h	DMA Channel 17 Virtual FIFO Read Pointer Register	DMA17_RDPTR
DMA + 0934h	DMA Channel 18 Virtual FIFO Read Pointer Register	DMA18_RDPTR
DMA + 09B4h	DMA Channel 19 Virtual FIFO Read Pointer Register	DMA19_RDPTR
DMA + 0A34h	DMA Channel 20 Virtual FIFO Read Pointer Register	DMA20_RDPTR
DMA + 0AB4h	DMA Channel 21 Virtual FIFO Read Pointer Register	DMA21_RDPTR
DMA + 0B34h	DMA Channel 22 Virtual FIFO Read Pointer Register	DMA22_RDPTR
DMA + 0BB4h	DMA Channel 23 Virtual FIFO Read Pointer Register	DMA23_RDPTR
DMA + 0C34h	DMA Channel 24 Virtual FIFO Read Pointer Register	DMA24_RDPTR

Table 227 DMA Virtual FIFO Read Pointer Registers List

DMA+08B8h DMA Channel 17 Virtual FIFO Data Count Register DMA17_FFCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FFCNT															
Type	RO															

The DMA Virtual FIFO Data Count registers for channel 17 to channel 24 are defined in registers listed in Table 23

FFCNT To display the number of data stored in FIFO. 0 means FIFO empty, and FIFO is full if FFCNT is equal to FFSIZE.

Register Address	Register Function	Acronym
DMA + 08B8h	DMA Channel 17 Virtual FIFO Data Count Register	DMA17_FFCNT
DMA + 0938h	DMA Channel 18 Virtual FIFO Data Count Register	DMA18_FFCNT
DMA + 09B8h	DMA Channel 19 Virtual FIFO Data Count Register	DMA19_FFCNT
DMA + 0A38h	DMA Channel 20 Virtual FIFO Data Count Register	DMA20_FFCNT
DMA + 0AB8h	DMA Channel 21 Virtual FIFO Data Count Register	DMA21_FFCNT
DMA + 0B38h	DMA Channel 22 Virtual FIFO Data Count Register	DMA22_FFCNT

DMA + 0BB8h	DMA Channel 23 Virtual FIFO Data Count Register	DMA23_FFCNT
DMA + 0C38h	DMA Channel 24 Virtual FIFO Data Count Register	DMA24_FFCNT

Table 23 DMA Virtual FIFO Data Count Registers List

DMA+08BCh DMA Channel 17 Virtual FIFO Status Register DMA17_FFSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**ALT**	**EMPTY**	**FULL**
Type														RO	RO	RO
Reset														0	1	0

The DMA Virtual FIFO Status registers for channel 17 to channel 24 are defined in registers listed in **Table 24**

FULL To indicate FIFO is full.

- **0** Not Full
- 1 Full

EMPTY To indicate FIFO is empty.

- **0** Not Empty
- 1 Empty

ALT To indicate FIFO Count is larger than ALTLEN. DMA issues an alert signal to UART to enable UART flow control.

- **0** Not reach alert region.
- 1 Reach alert region.

Register Address	**Register Function**	**Acronym**
DMA + 08BCh	DMA Channel 17 Virtual FIFO Status Register	DMA17_FFSTA
DMA + 093Ch	DMA Channel 18 Virtual FIFO Status Register	DMA18_FFSTA
DMA + 09BCh	DMA Channel 19 Virtual FIFO Status Register	DMA19_FFSTA
DMA + 0A3Ch	DMA Channel 20 Virtual FIFO Status Register	DMA20_FFSTA
DMA + 0ABCh	DMA Channel 21 Virtual FIFO Status Register	DMA21_FFSTA
DMA + 0B3Ch	DMA Channel 22 Virtual FIFO Status Register	DMA22_FFSTA
DMA + 0BBCh	DMA Channel 23 Virtual FIFO Status Register	DMA23_FFSTA
DMA + 0C3Ch	DMA Channel 24 Virtual FIFO Status Register	DMA24_FFSTA

Table 24 ***DMA Virtual FIFO Status Registers List***

DMA+08C0h DMA Channel 17 Virtual FIFO Alert Length Register DMA17_ALTLEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**ALTLEN**					
Type											R/W					
Reset											0					

The DMA Virtual FIFO Alert Length registers for channel 17 to channel 24 are defined in registers listed in **Table 25**

ALTLEN Specifies the Alert Length of Virtual FIFO DMA. Once the remaining FIFO space is less than ALTLEN, an alert signal is issued to UART to enable flow control. Normally, ALTLEN shall be larger than 16 for UART application.

Register Address	**Register Function**	**Acronym**
DMA + 08C0h	DMA Channel 17 Virtual FIFO Alert Length Register	DMA17_ALTLEN
DMA + 0940h	DMA Channel 18 Virtual FIFO Alert Length Register	DMA18_ALTLEN
DMA + 09C0h	DMA Channel 19 Virtual FIFO Alert Length Register	DMA19_ALTLEN
DMA + 0A40h	DMA Channel 20 Virtual FIFO Alert Length Register	DMA20_ALTLEN
DMA + 0AC0h	DMA Channel 21 Virtual FIFO Alert Length Register	DMA21_ALTLEN
DMA + 0B40h	DMA Channel 22 Virtual FIFO Alert Length Register	DMA22_ALTLEN
DMA + 0BC0h	DMA Channel 23 Virtual FIFO Alert Length Register	DMA23_ALTLEN
DMA + 0C40h	DMA Channel 24 Virtual FIFO Alert Length Register	DMA24_ALTLEN

Table 25 ***DMA Virtual FIFO Alert Length Registers List***

DMA+08C4h DMA Channel 17 Virtual FIFO Size Register DMA17_FFSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FFSIZE**															
Type	R/W															
Reset	0															

The DMA Virtual FIFO Size registers for channel 17 to channel 24 are defined in registers listed in **Table 26**

FFSIZE Specifies the FIFO Size of Virtual FIFO DMA.

Register Address	**Register Function**	**Acronym**
DMA + 08C4h	DMA Channel 17 Virtual FIFO Size Register	DMA17_FFSIZE
DMA + 0944h	DMA Channel 18 Virtual FIFO Size Register	DMA18_FFSIZE
DMA + 09C4h	DMA Channel 19 Virtual FIFO Size Register	DMA19_FFSIZE
DMA + 0A44h	DMA Channel 20 Virtual FIFO Size Register	DMA20_FFSIZE
DMA + 0AC4h	DMA Channel 21 Virtual FIFO Size Register	DMA21_FFSIZE

DMA + 0B44h	DMA Channel 22 Virtual FIFO Size Register	DMA22_FFSIZE
DMA + 0BC4h	DMA Channel 23 Virtual FIFO Size Register	DMA23_FFSIZE
DMA + 0C44h	DMA Channel 24 Virtual FIFO Size Register	DMA24_FFSIZE

Table 26 ***DMA Virtual FIFO Size Registers List***

2.7 AP CONFIG Register

2.7.1 APB Bridge Register Map

REGISTER ADDRESS	REGISTER NAME	SYNONYM
8000_1000h	Hardware Version Register	HW_VER
8000_1004h	Software Version Register	SW_VER
8000_1008h	Hardware Code Register	HW_CODE
8000_1010h	Software Misc. Low Register	SW_MISC_L
8000_1014h	Software Misc. High Register	SW_MISC_H
8000_1020h	Hardware Misc. Register	HW_MISC
8000_1100h	ARM9 Frequency Division Register	ARM9_FREQ_DIV
8000_1204h	Sleep Control Register	SLEEP_CON
8000_1208h	MCU Clock Control Register	MCUCLK_CON
8000_120Ch	EMI Clock Control Register	EMICLK_CON
8000_1300h	Subsystem Output Isolation Register	ISO_EN
8000_1304h	Subsystem Power Down Register	PWR_OFF
8000_1308h	APMCUSYS Memory Power Down Register	MCU_MEM_PDN
8000_130Ch	GRAPH1SYS Memory Power Down Register	G1_MEM_PDN
8000_1310h	GRAPH2SYS Memory Power Down Register	G2_MEM_PDN
8000_1314h	CEVASYS Memory Power Down Register	CEVA_MEM_PDN
8000_1318h	Subsystem Input Isolation Register	IN_ISO_EN
8000_131Ch	Subsystem Power Ack Register	PWR_ACK
8000_1320h	Subsystem Ack Clear Register	ACK_CLR
8000_1404h	APB Bus Control Register	APB_CON
8000_1408h	Security Boot Register	SECURITY_REG
8000_1500h	IO Driving Control Register 0	IO_DRV0
8000_1504h	IO Driving Control Register 1	IO_DRV1
8000_1600h	Instruction Cache Size Control Register	IC_SIZE
8000_1604h	Data Cache Size Control Register	DC_SIZE
8000_1608h	MDVCXO_OFF Register	MDVCXO_OFF

© 2009 MediaTek Inc.

2.7.2 Register Definitions

8000_1000h Hardware Version Register HW_VERSION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXTP				MAJREV				MINREV							
Type	RO				RO				RO				RO			
Reset	8				A				0				0			

This register is used by software to determine the hardware version of the chip. The register contains a new value whenever each metal fix or major step is performed. All values are incremented by a step of 1.

MINREV Minor Revision of the chip

MAJREV Major Revision of the chip

EXTP This field shows the existence of Hardware Code Register that presents the Hardware ID while the value is other than zero.

8000_1004h Software Version Register SW_VERSION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXTP				MAJREV				MINREV							
Type	RO				RO				RO				RO			
Reset	8				A				0				0			

This register is used by software to determine the software version used with this chip. All values are incremented by a step of 1.

MINREV Minor Revision of the software

MAJREV Major Revision of the software

EXTP This field shows the existence of Hardware Code Register that presents the Hardware ID when the value is other than zero.

8000_1008h Hardware Code Register HW_CODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CODE3				CODE2				CODE1				CODE0			
Type	RO				RO				RO				RO			
Reset	6				5				1				6			

This register presents the Hardware ID. CODE1 & CODE0 can be programmed by efuse_dout[61:54].

8000_1010h Software Misc Low Register SW_MISC_L

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SW_MISC_L															
Type	R/W															
Reset	0															

Spare registers for software control.

8000_1014h Software Misc High Register SW_MISC_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	SW_MISC_H
Type	R/W
Reset	0

Spare registers for software control.

8000_1020h Hardware Misc Register HW_MISC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		MD_BOOT_ONLY		NIRQ_MASK	MASK_GMC2	MASK_GMC1	CEVADBG_EN	NFI_SEL	SIM2_SEL	UART4_SEL	UART3_SEL	UART2_SEL	UART1_SEL	GMC_AUTOCG		USB_SEL
Type		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset		0		0	0	0	0	xcore	0	0	0	0	1	1		1

Spare registers for platform control.

USB_SEL USB selection.

- **0** AP use the USB.
- **1** MD use the USB.

GMC_AUTOCG HW automatic clock gating for GMC

- **0** Disable
- **1** Enable

UARTx_SEL UART selection.

- **0** AP use the UARTx.
- **1** MD use the UARTx.

SIM2_SEL SIM2 selection.

- **0** AP use the SIM2.
- **1** MD use the SIM2.

NFI_SEL This bit is used to set which domain NFI can access. AP and MD still can control the NFI no matter what you set this register

- **0** NFI can access MD domain.
- **1** NFI can access AP domain.

CEVADBG_EN CEVA debug request

- **0** Disable
- **1** Enable

MASK_GMC1 This bit is used to mask GMC clock gating bit in EMI slow idle condition

- **0** Un-mask GMC clock gating constraint. That is GMC clock must be gated before EMI slow down.
- **1** Mask GMC clock gating constraint. EMI can directly enter the slow idle mode without GMC clock gating constraint.

MASK_GMC2 This bit is used to mask GMC clock gating bit in EMI slow idle condition

- **0** Un-mask GMC clock gating constraint. That is GMC clock must be gated before EMI slow down.
- **1** Mask GMC clock gating constraint. EMI can directly enter the slow idle mode without GMC clock gating constraint.

NIRQ_MASK This bit is used to mask sleep controller’s wakeup signal, this signal is come from interrupt controller’s nIRQ signal.

- **0** Un-mask

1 Mask

MD_BOOT_ONLY This bit is used to let MD standalone rise PLL frequency without waking up AP side.

0 Disable

1 Enable this function

8000_1100h ARM9 Frequency Division Register ARM9_FREQ_DIV

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															ARM9_FREQ_DIV	
Type															R/W	
Reset															0	

ARM9_FREQ_DIV

00 ARM9 Clock is divided by 1, i.e. 416 MHz

01 Reserved

10 ARM9 Clock is divided by 2, i.e. 208 MHz

11 ARM9 Clock is divided by 4, i.e. 104 MHz

Note:

1. This register can be changed only if the source clock is switched to PLL.
2. The clock rate may not change immediately after the software sets this register. For any case which needs to assure that the clock really changes, you can read this register and wait until it becomes to the value that you specify.

8000_1204h Sleep Control Register SLEEP_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FMCU_2X_DIV_EN	FMCU_DIV_EN	F48M	MD_ACT_B	CEVA	DDR			AHB	
Type							R/W	R/W	R/W	R/W	R/W	R/W			R/W	
Reset							0	0	1	ICORE	1	0			0	

AHB Stop the AHB Bus Clock to force the entire bus to enter sleep mode. AHB clock will be resumed as long as there is an interrupt request or system is reset.

0 AHB Bus Clock is running

1 AHB Bus Clock is stopped

NOTE: Before entering bus sleep mode, you must ensure that CEVA, Graph1sys, and Graph2sys are not active.

DDR Stop the DDR Clock.

0 DDR Clock is running

1 DDR Clock is stopped

CEVA Stop the CEVA Clock.

0 CEVA Clock is stopped

1 CEVA Clock is running

MD_ACT_B Active MD MCU. The system boots from AP MCU by default (ICORE = 1). After AP MCU finish the initialization of the memory and system setting for MD MCU. AP MCU can set the bit as

"0" to activate MD MCU. The MD MCU will boot from MD_VECTOR. MD_VECTOR is specified by AP and the value is store in EMI. Whenever MD MCU access EMI, and address issued to external bus is ADDR + MD_VECTOR.

- **0** Active MD MCU
- **1** Disable MD MCU. (MD MCU clock stops)

F48M Stop the F48M Clock.

- **0** F48M Clock is running
- **1** F48M Clock is stopped

Note: Before switching the source clock to PLL, you should set both FMCU_DIV_EN and FMCU_2X_DIV_EN to 1 first, and then keep polling the registers until these two bits became 1. This procedure is a safe way to inform ARM9 to switch the clock ratio of CPU to BUS clocks, otherwise it might cause CPU crash.

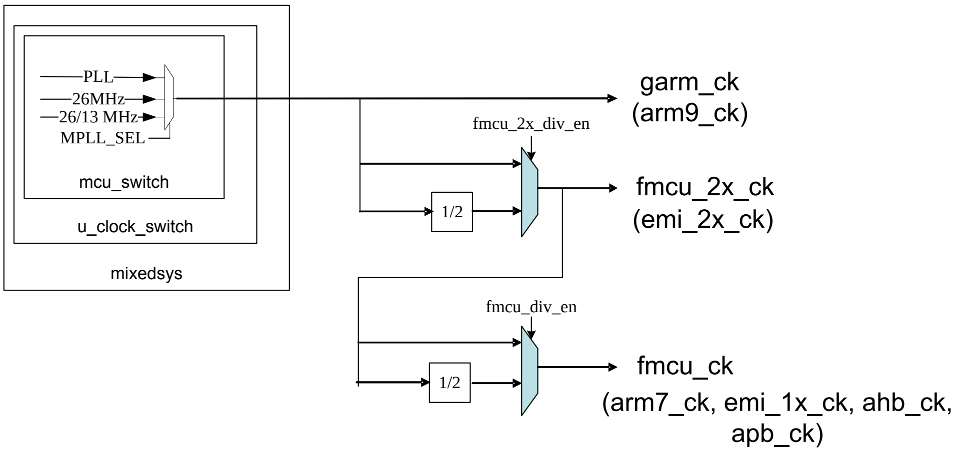


Figure 1 ***Clock Scheme***

FMCU_DIV_EN Enable fmcu_ck frequency division

- **0** Disable
- **1** Enable

FMCU_2X_DIV_EN Enable fmcu_2x_ck frequency division

- **0** Disable
- **1** Enable

8000_1208h MCU Clock Control Register MCUCLK_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													MCU_FSEL			
Type													R/W			
Reset													7			

MCU_FSEL MCU clock frequency selection. This control register is used to control the output clock frequency of MCU Dynamic Clock Manager. The clock frequency is from 13MHz to 104MHz. The waveforms of the output clock are shown below. This setting only takes effect when the bus has no any transaction and AHB bus clock has been stopped.

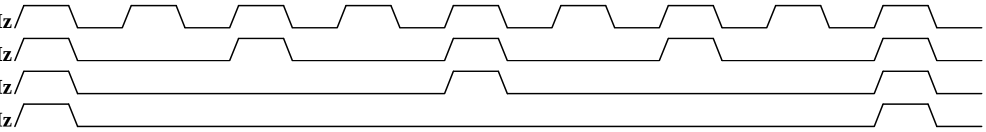


Figure 2 ***Output of Dynamic Clock Manager***

	High Speed Bus	Low Speed Bus
0	13MHz	13MHz
1	26MHz	26MHz
2	Reserved	Reserved
3	52MHz	52MHz
4	Reserved	Reserved
5	Reserved	Reserved
6	Reserved	Reserved
7	104MHz	52MHz
Others	Reserved	

8000_120Ch EMI Clock Control Register EMICLK_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**EMICLK_CON**				
Type												R/W				
Reset												0				

EMICLK_CON[4:0] EMI clock frequency selection

00000	3.25MHz
00001	6.5MHz
00011	13MHz
00111	26MHz
01111	52MHz
11111	104MHz
Others	Reserved

This register takes effect only when the following conditions are all true.

1. AP and MD AHB buses enter the sleep mode
2. GMC1, GMC2, and CEVA are all clock gating

NOTE: Before entering EMI slow idle mode, you must ensure that CEVA, Graph1sys, and Graph2sys are not active.

8000_1300h Subsystem Output Isolation Register ISO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**CEVA_ISO_EN**	**GRAPH2_ISO_EN**	**GRAPH1_ISO_EN**			
Type											R/W	R/W	R/W			
Reset											0	0	0			

Sub-system output isolation control

GRAPH1_ISO_EN Controls the graph1sys output signal isolation
GRAPH2_ISO_EN Controls the graph2sys output signal isolation
CEVA_ISO_EN Controls the cevasys output signal isolation

8000_1304h Subsystem Power Down Register PWR_OFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											CEVA_PDN	GRAPH2_PDN	GRAPH1_PDN			
Type											R/W	R/W	R/W			
Reset											0	0	0			

Sub-system power down control
GRAPH1_PDN Controls the graph1sys power down
GRAPH2_PDN Controls the graph2sys power down
CEVA_PDN Controls the cevasys power down

8000_1308h MCUSYS Memory Power Down Register MCU_MEM_PDN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													L1_TCM	L1_CACHE	MD_SYSROM	ETB
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CCIF	USB	AP_SYSROM	DTCM				ITCM				MMU	DC_16KB	DC	IC_16KB	IC
Type	R/W	R/W	R/W	R/W				R/W				R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0				0				0	0	0	0	0

MCUSYS memory power down control
IC Controls the lower instruction cache memory power down
IC_16KB Controls the upper instruction cache memory power down, when the cache configuration is 16KB, you should power down this memory block.
DC Controls the lower data cache memory power down
DC_16KB Controls the upper data cache memory power down, when the cache configuration is 16KB, you should power down this memory block.
MMU Controls the MMU memory power down
ITCM Controls the ITCM memory power down
DTCM Controls the DTCM memory power down
AP_SYSROM Controls the AP SYSROM power down
USB Contorl the USB memory power down
CCIF Control the CCIF memory power down
ETB Control the CSDBG memory power down
MD_SYSROM Controls the MD SYSROM power down
L1_CACHE Controls the L1 cache memory power down

L1_TCM Controls the L1 TCM memory power down

8000_130Ch GRAPH1SYS Memory Power Down Register G1_MEM_PDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							CAM	RESZ	IMGDMA	TVC	LCD	WAVE	ASM	AFE	DPI	DSI
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset							0	0	0	0	0	0	0	0	0	0

GRAPH1SYS memory power down control

DSI Controls the DSI memory power down

DPI Controls the DPI memory power down

AFE Controls the AFE memory power down

ASM Controls the ASM memory power down

WAVE Contorl the WAVE memory power down

LCD Control the LCD memory power down

TVC Control the TVC memory power down

IMGDMA Controls the IMGDMA memory power down

RESZ Controls the RESZ cache memory power down

CAM Controls the CAM memory power down

8000_1310h GRAPH2SYS Memory Power Down Register G2_MEM_PDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D
Type																R/W
Reset																0

GRAPH2SYS memory power down control

M3D Controls the M3D memory power down

8000_1314h CEVASYS Memory Power Down Register CEVA_MEM_PDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											CCIF	L2_MEM_PDN		L1_MEM_PDN		
Type											R/W	R/W		R/W		
Reset											0	0		0		

CEVASYS memory power down control

L1_MEM_DN Controls the L1 memory power down

L2_MEM_DN Controls the L2 memory power down

CCIF Controls the CCIF memory power down

8000_1318h Subsystem Input Isolation Register IN_ISO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											CEVA_IN_ISO	GRAPH2_IN_ISO	GRAPH1_IN_ISO			
Type											R/W	R/W	R/W			

Reset											0	0	0			

Sub-system output isolation control

GRAPH1_IN_ISO Controls the graph1sys input signal isolation

GRAPH2_IN_ISO Controls the graph2sys input signal isolation

CEVA_IN_ISO Controls the cevasys input signal isolation

8000_131Ch Subsystem Power Ack Register PWR_ACK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													MD2G_PWR_ACK	G1_PWR_ACK	G2_PWR_ACK	CEVA_PWR_ACK
Type													RO	RO	RO	RO
Reset													0	0	1	1

This register is used to indicate if the power down subsystem had powered up already.

8000_1320h Clear Subsystem Power Ack Register ACK_CLR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													MD2G_PWR_ACK	G1_PWR_ACK	G2_PWR_ACK	CEVA_PWR_ACK
Type													WO	WO	WO	WO
Reset													0	0	0	0

Note: Before using the power ack register to monitor the power ack of subsystems, you should clear powr ack registers first after powering down subsystems.
Writing to the corresponding "Clear" bit will perform a bit clear function.
Eg.
If PWR_ACK = 16'h000F,
Writing ACK_CLR = 16'000A will result in PWR_ACK = 16'h0005

8000_1404h APB Bus Control Register APB_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		APBW6	APBW5	APBW4	APBW3	APBW2	APBW1	APBW0		APBR6	APBR5	APBR4	APBR3	APBR2	APBR1	APBR0
Type		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0		0	0	0	0	0	0	0

This register is used to control the timing of Read Cycle and Write Cycle on APB Bus.

APBR0-APBR6 Read Access Time on APB Bus

- **0** 1-Cycle Access
- **1** 2-Cycle Access

APBW0-APBW6 Write Access Time on APB Bus

- **0** 1-Cycle Access
- **1** 2-Cycle Access

8000_1408h Security Boot Register SECURITY_BOOT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SECURITY_BOOT							
Type									R/W							
Reset									0							

This register is written by SW, and it is also readable for MD side.

8000_1500h IO Driving Control Register IO_DRV0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				CMPCLK	NFI				ETM CD				ETM CLOCK			
Type				R/W	R/W				R/W				R/W			
Reset				0	8				8				8			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM				DPI				PARALLEL LCD				SERIAL LCD			
Type	R/W				R/W				R/W				R/W			
Reset	8				8				8				8			

SERIAL LCD driving control of serial LCD IO
- **[1]** E4, add 4mA current
- **[2]** E2, add 2mA current
- **[3]** Slew rate control

PARALLEL LCD driving control of parallel LCD IO
- **[0]** E8, add 8mA current
- **[1]** E4, add 4mA current
- **[2]** E2, add 2mA current
- **[3]** Slew rate control

DPI driving control of DPI IO
- **[0]** E8, add 8mA current
- **[1]** E4, add 4mA current
- **[2]** E2, add 2mA current
- **[3]** Slew rate control

CAM driving control of camera IO
- **[0]** SMT control
- **[1]** E4, add 4mA current
- **[2]** E2, add 2mA current
- **[3]** Slew rate control

ETM CLOCK driving control of ETM clock
- **[0]** E8, add 8mA current
- **[1]** E4, add 4mA current
- **[2]** E2, add 2mA current
- **[3]** Slew rate control

ETM CD driving control of ETM control and data signal
- **[0]** E8, add 8mA current

[1] E4, add 4mA current
[2] E2, add 2mA current
[3] Slew rate control

NFI driving control of NFI IO
[1] E4, add 4mA current
[2] E2, add 2mA current
[3] Slew rate control

CMPCLK CMPCLK input SMT trigger control
0 disable
1 enable

8000_1504h IO Driving Control Register IO_DRV1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					I2C_2				I2C_1				I2C_0			
Type					R/W				R/W				R/W			
Reset					8				8				8			

I2C_0 driving control of I2C_0
[0] E8, add 8mA current
[1] E4, add 4mA current
[2] E2, add 2mA current
[3] Slew rate control

I2C_1 driving control of I2C_1
[0] E8, add 8mA current
[1] E4, add 4mA current
[2] E2, add 2mA current
[3] Slew rate control

I2C_2 driving control of I2C_2
[0] E8, add 8mA current
[1] E4, add 4mA current
[2] E2, add 2mA current
[3] Slew rate control

8000_1600h ARM926EJS Instruction Cache Size Register IC_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																IC_SIZE
Type																R/W
Reset																0

This register is used to configure the instruction cache size of ARM926EJ-S.

IC_SIZE
0 32KB cache size
1 16KB cache size

© 2009 MediaTek Inc.

8000_1604h ARM926EJS Data Cache Size Register DC_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DC_SIZE
Type																R/W
Reset																0

This register is used to configure the data cache size of ARM926EJ-S.

DC_SIZE

0 32KB cache size

1 16KB cache size

8000_1608h MDVCXO_OFF Register MDVCXO_OFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																MDVCXO_OFF
Type																RO
Reset																mdvcxo_off

This register is used to monitor mdvcxo_off signal

2.7.3 MTCMOS Application Note

MT6516 implements MTCMOS technology. The following subsystem can be powered off when no use:

1) GRAPH1SYS

2) GRAPH2SYS

3) CEVASYS

4) MD2GSYS

The following figure lists all power regions. The gray region in figure can be powered down by proper procedure.

The following table summarizes related control registers and power-on stable time.

The following list summarizes which sections you can find the related MTCMOS control registers.

AP CONFIG Register: IN_ISO_EN(AP), ISO_EN(AP), PWR_OFF(AP), and SLEEP_CON(AP).

MD CONFIG Register: IN_ISO_EN(MD), ISO_EN(MD), PWR_CON(MD), and SLEEP_CON(MD)

Reset Generation Unit(APRGU): RGU_USRST2, RGU_USRST3, RGU_USRST4, and RGU_USRST5

GRAPH1SYS CONFIG Register: GRAPH1SYS_CG_SET and GRAPH1SYS_CG_CLR

GRAPH2SYS CONFIG Register: GRAPH2SYS_CG_SET and GRAPH2SYS_CG_CLR

For safely powering on/down each subsystem, the following statements are our proposed power on/down procedure.

- Power-on sequence
  - 1) Enable subsystem software reset
  - 2) Power-on subsystem

- 3) Wait for a power-on stable time
- 4) Disable input isolation
- 5) Enable clocks
- 6) Disable output isolation
- 7) Disable subsystem software reset

- Power-down sequence
  - 1) Enable output isolation
  - 2) Disable clocks
  - 3) Enable input isolation
  - 4) Power-down subsystem

2.8 APMCUSYS CONFIG Register

In addition to the Pause Mode capability while in the Standby State, the software program can also put each peripheral independently into Power Down Mode while in the Active State by gating off their clock. The typical logic implementation is depicted in **Figure 13**. For all configuration bits, 1 signifies that the function is in Power Down Mode, and 0 means the function is in the Active Mode.

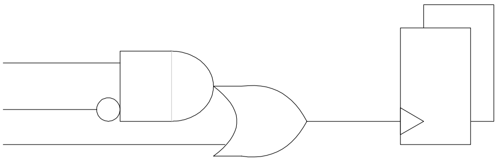


Figure 13 ***Power Down Control at Block Level***

REGISTER ADDRESS	REGISTER NAME	SYNONYM
8003_9300h	Clock Gating Control Status Register 0	APMCUSYS_PDN_CON0
8003_9320h	Clock Gating Set Register 0	APMCUSYS_PDN_SET0
8003_9340h	Clock Gating Clear Register 0	APMCUSYS_PDN_CLR0
8003_9360h	Clock Gating Control Status Register 1	APMCUSYS_PDN_CON1
8003_9380h	Clock Gating Set Register 1	APMCUSYS_PDN_SET1
8003_93A0h	Clock Gating Clear Register 1	APMCUSYS_PDN_CLR1
8003_9600h	Memory Delsel Control Register 0 (Used by Hardware)	APMCUSYS_DELSEL0
8003_9604h	Memory Delsel Control Register 1 (Used by Hardware)	APMCUSYS_DELSEL1
8003_9608h	Memory Delsel Control Register 2 (Used by	APMCUSYS_DELSEL2

	Hardware)	
8003_960Ch	Memory Delsel Control Register 3 (Used by Hardware)	APMCUSYS_DELSEL3
8003_9700h	ARM9 Monitor Control Register	APMCUSYS_MON_CON
8003_9704h	ARM9 Monitor Set Register	APMCUSYS_MON_SET
8003_9708h	ARM9 Monitor Clear Register	APMCUSYS_MON_CLR
8003_970Ch	ARM9 Performance Register 1	APMCUSYS_MON_PERF1
8003_9710h	ARM9 Performance Register 2	APMCUSYS_MON_PERF2
8003_9714h	ARM9 Performance Register 3	APMCUSYS_MON_PERF3
8003_9718h	ARM9 Performance Register 4	APMCUSYS_MON_PERF4
8003_971Ch	ARM9 Performance Register 5	APMCUSYS_MON_PERF5
8003_9720h	ARM9 Performance Register 6	APMCUSYS_MON_PERF6
8003_9724h	ARM9 Performance Register 7	APMCUSYS_MON_PERF7
8003_9728h	ARM9 Performance Register 8	APMCUSYS_MON_PERF8
8003_972Ch	ARM9 Performance Register 9	APMCUSYS_MON_PERF9
8003_9730h	ARM9 Performance Register 0	APMCUSYS_MON_PERF10
8003_9734h	ARM9 Performance Register 11	APMCUSYS_MON_PERF11
8003_9738h	ARM9 Performance Register 12	APMCUSYS_MON_PERF12
8003_973Ch	ARM9 Performance Register 13	APMCUSYS_MON_PERF13
8003_9740h	ARM9 Performance Register 14	APMCUSYS_MON_PERF14
8003_9744h	ARM9 Performance Register 15	APMCUSYS_MON_PERF15
8003_9748h	ARM9 Performance Register 16	APMCUSYS_MON_PERF16
8003_974Ch	ARM9 Performance Register 17	APMCUSYS_MON_PERF17
8003_9750h	ARM9 Performance Register 18	APMCUSYS_MON_PERF18
8003_9754h	ARM9 Performance Register 19	APMCUSYS_MON_PERF19
8003_9758h	ARM9 Performance Register 20	APMCUSYS_MON_PERF20
8003_975Ch	ARM9 Performance Register 21	APMCUSYS_MON_PERF21
8003_9760h	ARM9 Performance Register 22	APMCUSYS_MON_PERF22

Table 27 ***APB Bridge Register Map***

2.8.1 Register Definitions

8003_9300h Clock Gating Control Status Register 0 APMCUSYS_PDN_CON0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ONEWIRE	MSDC3	UART4	XGPT	TP	AUXADC			MSDC2	SIM2	I2C		IRDA	I2C2	NFI	SWDBG
Type	RO	RO	RO	RO	RO	RO			RO	RO	RO		RO	RO	RO	RO
Reset	1	1	0	1	1	1			1	1	1		1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

© 2009 MediaTek Inc.

Name	MSDC	PWM3	PWM2	PWM1	PWM	SIM	UART3	UART2	UART1	GPIO	KP	GPT	I2C3	SEJ	USB	DMA
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0

APMCU sub-system power down control status register (read only). Value 1 represents power down.

DMA Status of the AP DMA Controller Power Down

USB Status of the USB Power Down

SEJ Status of the SEJ Power Down.

I2C3 Status of the 3rd I2C Controller Power Down

GPT Status of the GPT timer Power Down.

KP Status of the keypad Power Down.

GPIO Status of the GPIO Power Down.

UART1 Status of the 1st UART Power Down.

UART2 Status of the 2nd UART Power Down.

UART3 Status of the 3rd UART Power Down.

SIM Status of the 1st SIM Power Down.

PWM Status of PWM module Power Down. Set this bit to 1 would power down all 7 PWM (PWM0, PWM1,…,PWM6).

PWM0 Status of the PWM0 Power Down.

PWM1 Status of the PWM1 Power Down.

PWM2 Status of the PWM2 Power Down.

MSDC Status of the 1st MSDC Power Down.

SWDBG Status of the Software Debug Power Down.

NFI Status of the NFI Power Down.

I2C2 Status of the 2nd I2C Power Down.

IRDA Status of the IRDA Power Down.

I2C Status of the 1st I2C Power Down.

SIM2 Status of the 2nd SIM Power Down.

MSDC2 Status of the 2nd MSDC Power Down.

AUXADC Status of the AUXADC Power Down.

TP Status of the Touch Panel Power Down.

XGPT Status of the XGPT timer Power Down.

UART4 Status of the 4th UART Power Down.

MSDC3 Status of the 3rd MSDC Power Down.

ONEWIRE Status of the ONEWIRE Power Down.

8003_9320h Clock Gating Control Set Register 0 APMCUSYS_PDN_SET0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ONEWIRE	MSDC3	UART4	XGPT	TP	AUXADC			MSDC2	SIM2	I2C		IRDA	I2C2	NFI	SWDBG
Type	WO	WO	WO	WO	WO	WO			WO	WO	WO		WO	WO	WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5		3	2	1	0
Name	MSDC	PWM3	PWM2	PWM1	PWM	SIM	UART3	UART2	UART1	GPIO	KP	GPT	I2C3	SEJ	USB	DMA

Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

APMCU sub-system power down set register, value 1 represents power down. For all registers addresses listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16’h0F0F, writing PDN_SET0 = 16’F0F0 will result in PDN_CON0 = 16’hFFFF.

8003_9340h Clock Gating Control Clear Register 0 APMCUSYS_PDN_CLR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ONEWIRE**	**MSDC3**	**UART4**	**XGPT**	**TP**	**AUXADC**			**MSDC2**	**SIM2**	**I2C**		**IRDA**	**I2C2**	**NFI**	**SWDBG**
Type	WO	WO	WO	WO	WO	WO			WO	WO	WO		WO	WO	WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5		3	2	1	0
Name	**MSDC**	**PWM3**	**PWM2**	**PWM1**	**PWM**		**UART3**	**UART2**	**UART1**	**GPIO**	**KP**	**GPT**	**I2C3**	**SEJ**	**USB**	**DMA**
Type	WO	WO	WO	WO	WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

APMCU sub-system power down clear register, value 1 represents power up. For all registers addresses listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16’hFFFF, writing PDN_CLR0 = 16’F0F0 will result in PDN_CON0 = 16’h0F0F.
USB Please be noticed that there’s programming constraints for USB. The difference between USB and others is USB has its registers accessed through AHB instead of APB. What makes this fact serious is the power up operation and USB register access are two distinct paths in hardware, the APB and the AHB. We must first power up the USB to un-gate its AHB clock otherwise we cannot access its registers. Then look at the example below:

Instruction 1: APB write to power up USB
Instruction 2: AHB access to USB register
Instruction 3: …

The instruction 1 is to un-gate the AHB clock of the USB. There’re chances that the un-gated AHB clock has not yet propagated to USB when instruction 2 arrives USB and consequently the register access of instruction 2 failed. To get rid of the potential problem, we suggested the below programming codes:

Instruction 1: APB write to power up USB
Loop 1-a:Read USB register 0x8010061B and loop until the return value is 0x80
Instruction 2: AHB access to USB register
Instruction 3: …

The Loop 1-a guarantees that the clock propagation reaches USB because the returned data will be zeros if the AHB clock of USB is still gated.

8003_9360h Clock Gating Control Status Register 1 APMCUSYS_ PDN_CON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name															**PWM0**	**CSDBG**
Type															RO	RO
Reset															1	1

APMCU sub-system power down direct status, value 1 represents power down.

CSDBG Status of the CSDBG Power Down. This status takes effect immediately.

PWM0 Status of the PWM0 Power Down. This status takes effect immediately.

8003_9380h Clock Gating Set Register 1 APMCUSYS_PDN_SET1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**PWM0**	**CSDBG**
Type															WO	WO

APMCU sub-system power down set register, value 1 represents power down. For all registers addresses listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16’h0F0F, writing PDN_SET0 = 16’F0F0 will result in PDN_CON0 = 16’hFFFF.

8003_93A0h Clock Gating Clear Register 1 APMCUSYS_PDN_CLR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**PWM0**	**CSDBG**
Type															WO	WO

MDMCU sub-system power down clear register, value 1 represents power up. For all registers addresses listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16’hFFFF, writing PDN_CLR0 = 16’F0F0 will result in PDN_CON0 = 16’h0F0F.

8003_9600h Memory Delsel Control Register 0 APMCUSYS_DELSEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARM9**															
Type	RW															
Reset	01		01		01		01		01		01		01		01	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**USB**				**ETB**				**CCIF**				**CCIF_CEVA**			
Type	RW				R/W				R/W				R/W			
Reset	0011				0001				0010				0001			

8003_9604h- 8003_9608h Memory Delsel Control Register 1-2 APMCUSYS_DELSEL1 - APMCUSYS_DELSEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARM9															
Type	RW															
Reset	01		01		01		01		01		01		01		01	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARM9															
Type	RW															
Reset	01		01		01		01		01		01		01		01	

8003_960Ch Memory Delsel Control Register 3 APMCUSYS_DELSEL21

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CSDBG		APSYSROM			
Type											RW		RW			
Reset											10		11		00	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	APSYSROM				ARM9											
Type	RW				RW											
Reset	11		10		01		01		01		01		01		01	

8003_9700h ARM9 Monitor Control Register APMCUSYS_MON_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									DAHB_SEL		IAHB_CLR	DAHB_CLR	IEXT_CLR	DEXT_CLR	ICP_CLR	DCP_CLR
Type									RO		RO	RO	RO	RO	RO	RO
Reset									0		1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ITLB_CLR	DTLB_CLR	ACTIVE_CLR	ICM_CLR	DCM_CLR	IAHB_EN	DAHB_EN	IEXT_EN	DEXT_EN	ICP_EN	DCP_EN	ITLB_EN	DTLB_EN	ACTIVE_EN	ICM_EN	DCM_EN
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

DCM_EN Enable the miss rate monitor of data cache

ICM_EN Enable the miss rate monitor of instruction cache

ACTIVE_EN Enable ARM9 active counter

DTLB_EN Enable the data TLB penalty counter of MMU

ITLB_EN Enable the instruction TLB penalty counter of MMU

DCP_EN Enable the penalty counter of data cache

ICP_EN Enable the penalty counter of instruction cache

DEXT_EN Enable the penalty counter of data external write buffer

IEXT_EN	Enable the penalty counter of instruction external write buffer
DAHB_EN	Enable the penalty counter of ARM9 data AHB bus
IAHB_EN	Enable the penalty counter of ARM9 instruction AHB bus
DCM_CLR	Clear the miss rate counter of data cache (active low)
ICM_CLR	Clear the miss rate counter of instruction cache (active low)
ACTIVE_CLR	Clear ARM9 active counter (active low)
DTLB_CLR	Clear the data TLB penalty counter of MMU (active low)
ITLB_CLR	Clear the instruction TLB penalty counter of MMU (active low)
DCP_CLR	Clear the penalty counter of data cache (active low)
ICP_CLR	Clear the penalty counter of instruction cache (active low)
DEXT_CLR	Clear the penalty counter of data external write buffer (active low)
IEXT_CLR	Clear the penalty counter of instruction external write buffer (active low)
DAHB_CLR	Clear the penalty counter of data AHB bus (active low)
IAHB_CLR	Clear the penalty counter of instruction AHB bus (active low)
DAHB_SEL	This control register is used to select which the data address range is monitored. It is taken effect when you enable the penalty counter of ARM9 data AHB bus.

0	External memory: 0x0000_0000 ~ 0x3FFF_FFFF
1	Internal memory: 0x4000_0000 ~ 0x4FFF_FFFF
2	APB register: 0x8000_0000 ~ 0xFFFF_FFFF
3	Reserved

8003_9704h ARM9 Monitor Set Register APMCUSYS_MON_SET

For monitor control register listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding MON_CON register.

Eg.

If MON_CON = 16’h0F0F,

Writing MON_SET = 16’F0F0 will result in MON_CON = 16’hFFFF.

8003_9708h ARM9 Monitor Clear Register APMCUSYS_MON_CLR

For monitor control register listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding MON_CON register.

Eg.

If MON_CON = 16’hFFFF,

Writing MON_CLR = 16’F0F0 will result in MON_CON = 16’h0F0F.

8003_970Ch ARM9 Performance register 1 APMCUSYS_MON_PERF1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name	DC_READ_REQ															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DC_READ_REQ															
Type	RO															
Reset	0															

8003_9710h ARM9 Performance register 2 APMCUSYS_MON_PERF2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DC_WRITE_REQ															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DC_WRITE_REQ								DC_READ_REQ							
Type	RO								RO							
Reset	0								0							

8003_9714h ARM9 Performance register 3 APMCUSYS_MON_PERF3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DC_READ_MISS															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DC_WRITE_REQ															
Type	RO															
Reset	0															

8003_9718h ARM9 Performance register 4 APMCUSYS_MON_PERF4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DC_WRITE_MISS								DC_READ_MISS							
Type	RO								RO							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DC_READ_MISS															
Type	RO															
Reset	0															

8003_971Ch ARM9 Performance register 5 APMCUSYS_MON_PERF5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DC_WRITE_MISS															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DC_WRITE_MISS															
Type	RO															
Reset	0															

8003_9720h ARM9 Performance register 6 APMCUSYS_MON_PERF6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IC_READ_REQ**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IC_READ_REQ**															
Type	RO															
Reset	0															

8003_9724h ARM9 Performance register 7 APMCUSYS_MON_PERF7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IC_READ_MISS**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IC_READ_MISS**								**IC_READ_REQ**							
Type	RO								RO							
Reset	0								0							

8003_9728h ARM9 Performance register 8 APMCUSYS_MON_PERF8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARM9_ACTIVE**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IC_READ_MISS**															
Type	RO															
Reset	0															

8003_972Ch ARM9 Performance register 9 APMCUSYS_MON_PERF9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DTLB_PENALTY**								**ARM9_ACTIVE**							
Type	RO								RO							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ARM9_ACTIVE**															
Type	RO															
Reset	0															

8003_9730h ARM9 Performance register 10 APMCUSYS_MON_PERF10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DTLB_PENALTY**															
Type	RO															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DTLB_PENALTY**															
Type	RO															
Reset	0															

8003_9734h ARM9 Performance register 11 APMCUSYS_MON_PERF11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ITLB_PENALTY**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ITLB_PENALTY**															
Type	RO															
Reset	0															

8003_9738h ARM9 Performance register 12 APMCUSYS_MON_PERF12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DC_PENALTY**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DC_PENALTY**								**ITLB_PENALTY**							
Type	RO								RO							
Reset	0								0							

8003_973Ch ARM9 Performance register 13 APMCUSYS_MON_PERF13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IC_PENALTY**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DC_PENALTY**															
Type	RO															
Reset	0															

8003_9740h ARM9 Performance register 14 APMCUSYS_MON_PERF14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEXT_PENALTY**								**IC_PENALTY**							
Type	RO								RO							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IC_PENALTY**															
Type	RO															
Reset	0															

8003_9744h ARM9 Performance register 15 APMCUSYS_MON_PERF15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEXT_PENALTY															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEXT_PENALTY															
Type	RO															
Reset	0															

8003_9748h ARM9 Performance register 16 APMCUSYS_MON_PERF16

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IEXT_PENALTY															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IEXT_PENALTY															
Type	RO															
Reset	0															

8003_974Ch ARM9 Performance register 17 APMCUSYS_MON_PERF17

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAHB_PENALTY															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAHB_PENALTY								IEXT_PENALTY							
Type	RO								RO							
Reset	0								0							

8003_9750h ARM9 Performance register 18 APMCUSYS_MON_PERF18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAHB_REQ															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAHB_PENALTY															
Type	RO															
Reset	0															

8003_9754h ARM9 Performance register 19 APMCUSYS_MON_PERF19

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IAHB_PENALTY								DAHB_REQ							
Type	RO								RO							

Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DAHB_REQ**															
Type	RO															
Reset	0															

8003_9758h ARM9 Performance register 20 APMCUSYS_MON_PERF20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IAHB_PENALTY**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IAHB_PENALTY**															
Type	RO															
Reset	0															

8003_975Ch ARM9 Performance register 21 APMCUSYS_MON_PERF21

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IAHB_REQ**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IAHB_REQ**															
Type	RO															
Reset	0															

8003_9760h ARM9 Performance register 22 APMCUSYS_MON_PERF22

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**IAHB_REQ**							
Type									RO							
Reset									0							

For all register addresses listed above, each counter register is 40-bit width. The 40-bit monitor can record about 43.98 minutes when ARM9 runs at 416MHz, that is “clock period x $2^{\text{counter bit width}}$ = 2.4ns x 2^{40} = 43.98 mins”.

DC_READ_REQ Total read requests of the data cache

DC_WRITE_REQ Total write requests of the data cache

DC_READ_MISS Total read misses of the data cache, read miss rate = total read misses / total read requests.

DC_WRITE_MISS Total write misses of the data cache, write miss rate = total write misses / total write requests.

IC_READ_REQ Total read requests of the instruction cache

IC_READ_MISS Total read misses of the instruction cache, read miss rate = total read misses / total read requests.
ARM9_ACTIVE ARM9 total active count, it only counts when ARM9 is active.
DTLB_PENALTY It counts ARM9 stall cycles caused by data TLB.
ITLB_PENALTY It counts ARM9 stall cycles caused by instruction TLB.
DC_PENALTY It counts ARM9 stall cycles caused by data cache.
IC_PENALTY It counts ARM9 stall cycles caused by instruction cache.
DEXT_PENALTY It counts ARM9 stall cycles caused by data external write buffer (DEXT).
IEXT_PENALTY It counts ARM9 stall cycles caused by instruction external write buffer (IEXT)
DAHB_PENALTY This counter will count the penalty caused by the external bus (Data AHB bus), and this counter will be influenced by the DAHB_SEL register, if you set DAHB_SEL as 0x1, the penalty monitored by the counter will only has the internal memory access penalty not the total penalty. With this functionality, we can further analyze the penalty source with the same monitor.
DAHB_REQ Total request number of the ARM9 data bus.
IAHB_PENALTY This counter will count the penalty caused by the external bus (Instruction AHB bus).
IAHB_REQ Total request number of the ARM9 instruction bus.

2.9 AP eXtended GPT

2.9.1 General Description

AP-domain Extended General Purpose Timer (APXGPT). Channels 1 & 2 are based on 13,000,000Hz (13MHz) clock and channels 3 to 7 are based on 32768Hz clock.

2.9.2 Register Definitions

XGPT+0000h APXGPT IRQ enable XGPT_IRQEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

IENn Enable the interrupt of each XGPT channel. When the COUNTER is equal to COMPARE and mode is not FREERUN.
- **0** Interrupt of channel n is disabled.
- **1** Interrupt of channel n is enabled.

XGPT+0004h APXGPT IRQ status XGPT_IRQSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**IST7**	**IST6**	**IST5**	**IST4**	**IST3**	**IST2**	**IST1**
Type										RO	RO	RO	RO	RO	RO	RO
Reset																

ISTn Interrupt status for channel n

- **0** No interrupt is generated
- **1** An interrupt is pending and waiting for service

XGPT+0008h APXGPT IRQ status acknowledge XGPT_IRQACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**IACK7**	**IACK6**	**IACK5**	**IACK4**	**IACK3**	**IACK2**	**IACK1**
Type										WO	WO	WO	WO	WO	WO	WO
Reset																

IACKn Interrupt acknowledge for the APXGPT channel

- **0** No effect
- **1** Interrupt request is acknowledged and should be relinquished.

XGPT+00n0h APXGPT channel n control XGPTn_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**MODE**				**CLR**	**EN**
Type											R/W				WO	R/W
Reset											0					0

n=1 to 7

EN Enable the XGPT channel n.

- **0** XGPT channel n is disabled.
- **1** XGPT channel n is enabled.

CLR Clear the COUNTn to 0.

MODE The operation mode of channel n

- **00** ONE-SHOT mode.
- **01** REPEAT mode.
- **10** KEEPGO mode.
- **11** FREERUN mode.

mode	Auto stop	Interrupt	Increase when EN=1 and …	If COUNTn = COMPAREn	Example: reset to 0 and compare = 2 (**Bold means interrupt**.)
ONE-SHOT	Yes	Yes	Stopped when COUNTn equals to COMPAREn	EN is reset to 0	0,1,**2**,2,2,2,2,2,2,2,2,…
REPEAT	No	Yes		Count is reset to 0	0,1,**2**,0,1,**2**,0,1,**2**,0,1,**2**,0,….
KEEPGO	No	Yes	Reset to 0 when overflow		0,1,**2**,3,4,5,6,7,8,9,…
FREERUN	No	No	Reset to 0 when overflow		0,1,2,3,4,5,6,7,8,9,…

XGPT+00n4h APXGPT channel n prescaler XGPTn_PRESCALER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name														**PRESCALER**		
Type														R/W		
Reset														0		

n=1 to 7

PRESCALE XGPT channel n input clock.

Clock (Hz)	channel	
prescale	1 to 2	3 to 7
000	13M	32768
001	6.5M	16384
010	3.25M	8192
011	1.625M	4096
100	812.5K	2048
101	406.25K	1024
110	203.125K	512
111	101.5625K	256

Note: M=1,000,000; K=1,000.

XGPT+00n8h APXGPT channel n count XGPTn_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COUNT[31:16]**															
Type	RO															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**COUNT[15:0]**															
Type	RO															
Reset	0															

n=1 to 7

COUNT The current count of channel n. When EN=1, the COUNT increases according prescaler.

XGPT+00nch APXGPT channel n compare value XGPTn_COMPARE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMPARE[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMPARE[15:0]															
Type	R/W															
Reset	0															

n=1 to 7

COMPARE The compared value of channel n. When (EN=1) and (MODE is not FREERUN) and (COUNT is equal to COMPARE) and (IEN=1), an interrupt happened.

2.10 Auxiliary ADC Unit

The auxiliary ADC unit is used to monitor the status of the battery and charger, to identify the plugged peripheral, and to perform temperature measurement. 9 input channels allow diverse applications in this unit.

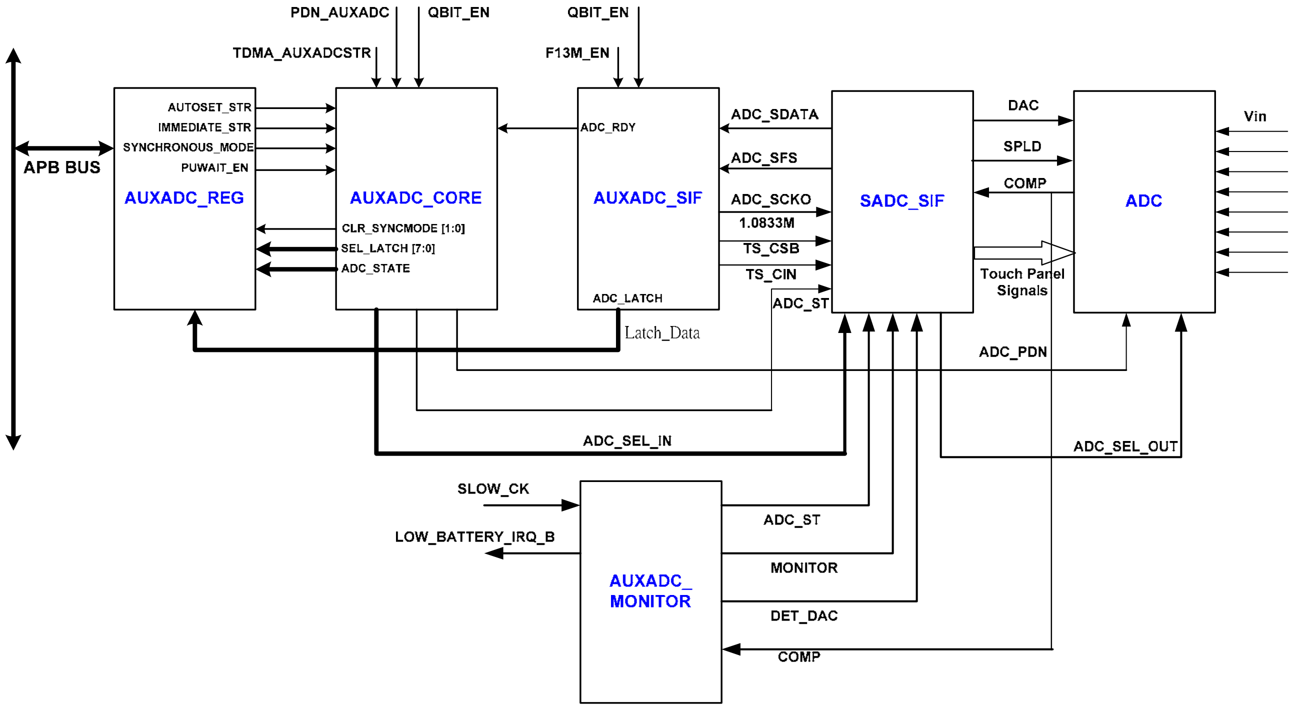


Figure 14 Auxadc Architecture

Each channel can operate in one of two modes: immediate mode and timer-triggered mode. The mode of each channel can be individually selected through register AUXADC_CON0. For example, if the flag SYN0 in

the register AUXADC_CON0 is set, Channel 0 is set in the timer-triggered mode. Otherwise, the channel operates in the immediate mode.

In the immediate mode, the A/D converter samples the value once only when the flag in the AUXADC_CON1 register has been set. For example, if the flag IMM0 in AUXADC_CON1 is set, the A/D converter samples the data for Channel 0. The IMM flags must be cleared and set again to initiate another sampling.

The value sampled for Channel 0 is stored in register AUXADC_DAT0, the value for Channel 1 is stored in register AUXADC_DAT1, etc.

If the AUTOSET flag in the register AUXADC_CON3 is set, the auto-sampling function is enabled. The A/D converter samples the data for the channel on which the corresponding data register has been read. For example, in the case where the SYN1 flag is not set, the AUTOSET flag is set, when the data register AUXADC_DAT0 has been read, the A/D converter samples the next value for Channel 1 immediately.

If multiple channels are selected at the same time, the task is performed sequentially on every selected channel. For example, if AUXADC_CON1 is set to 0x1ff, that is, all the 9 channels are selected, the state machine in the unit starts sampling from Channel 8 to Channel 0, and saves the values of each input channel in its corresponding register. The same process also applies to the timer-triggered mode.

In the timer-triggered mode, the A/D converter samples the value for the channels in which the corresponding SYN flags are set when the TDMA timer counts to the value specified in the register TDMA_AUXEV1, which is placed in the TDMA timer. For example, if AUXADC_CON0 is set to 0x1ff, all the 9 channels are selected the timer-triggered mode. The state machine samples all the 9 channels sequentially and saves the values in registers from AUXADC_DAT0 to AUXADC_DAT8, as it does in the immediate mode.

There is a dedicated timer-triggered scheme for Channel 0. This scheme is enabled by setting the SYN9 flag in the register AUXADC_CON2. The timing offset for this event is stored in the register TDMA_AUXEV0 in the TDMA timer. The sampled data triggered by this specific event is stored in the register AUXADC_DAT9. It is used to separate the results of two individual software routines that perform actions on the auxiliary ADC unit.

The AUTOCLR*n* in the register AUXADC_CON3 is set when it is intended to sample only once after setting the timer-triggered mode. If AUTOCLR1 flag has been set, after the data for the channels in the timer-triggered mode has been stored, the SYN*n* flags in the register AUXADC_CON0 are cleared. If AUTOCLR0 flag has been set, after the data for the channel 0 has been stored in the register AUXADC_DAT9, the SYN9 flag in the register AUXADC_CON2 is cleared.

The usage of the immediate mode and timer-triggered mode are mutually exclusive in each individual channel.

The PUWAIT_EN bit in the registers AUXADC_CON3 is used to power up the analog port in advance. This ensures that the power has ramped up to the stable state before A/D converter starts the conversion. The analog part is automatically powered down after the conversion is completed.

Touch Panel:

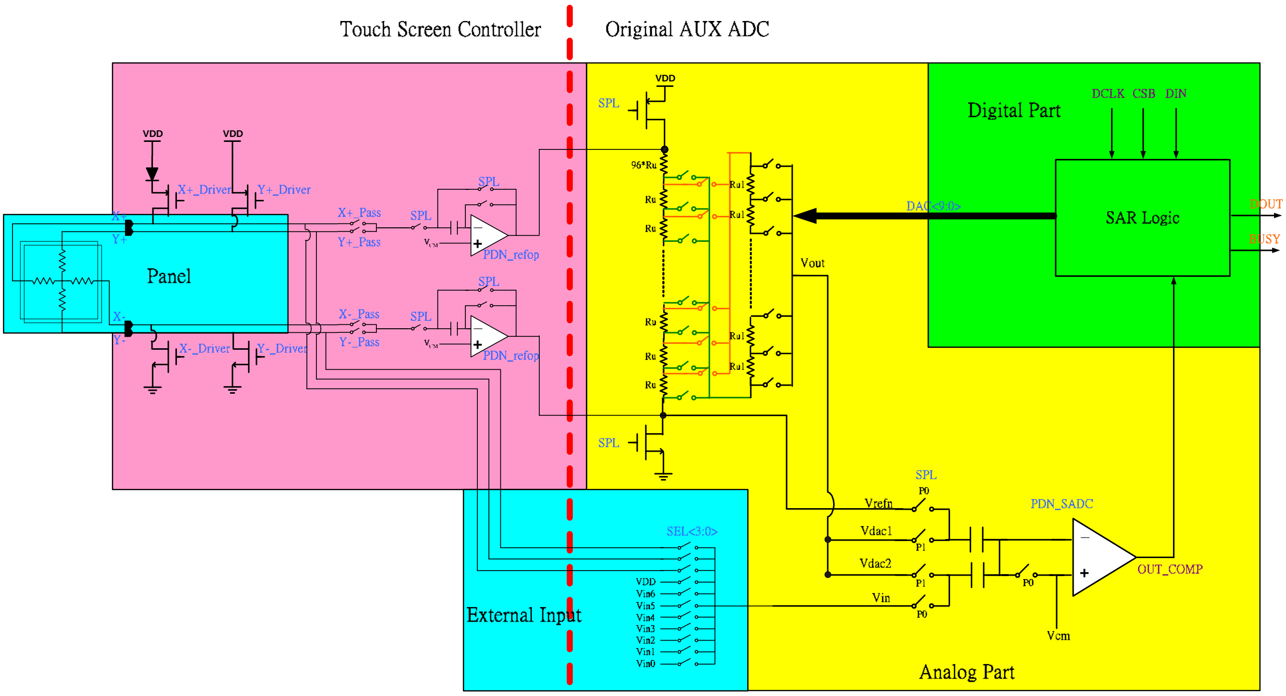


Figure 15 Touch Panel Circuit Structure

Besides the normal sampling of the external input voltage, auxadc includes the sampling of the touch panel function. For the specified axis, SW should program **AUX_TS_CMD** first, and then trigger touch panel's sample in the register **AUX_TS_CON**. The touch panel sampling waveform is shown as follows. After SW polls status bit in the register **AUXADC_CON3** to know that the touch panel sample is finished. SW can read back the specified axis value from the register **AUX_TS_DAT0**.

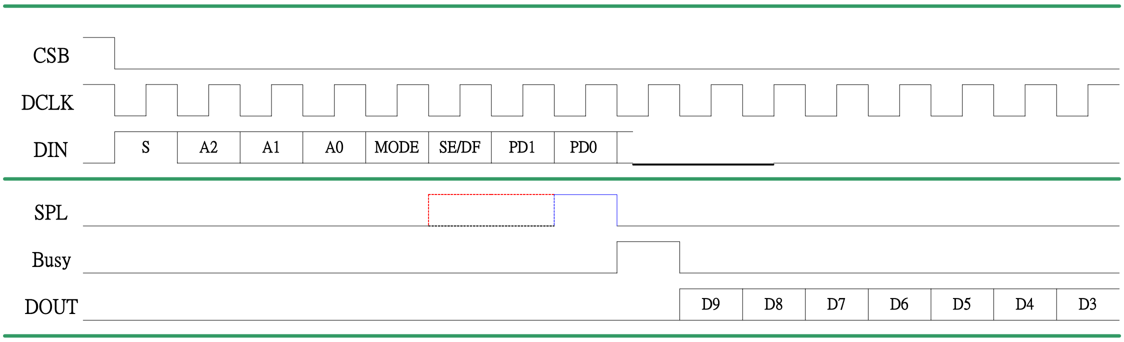


Figure 16 Touch Panel Sampling Waveform

S: Start bit

A2~A0: Addressing bits

Mode: 10-bit or 8-bit

SE/DF: Single End or Differential mode

 © 2009 MediaTek Inc.

PD1~0: Power Down Command

These values are defined in the register **AUX_TS_CMD**. In the following table, it shows the relationship between AUX_TS_CMD and touch panel control signals.

SE/DFB	A2	A1	A0	X/Y Driver	X/Y Pass	SEL< 3:0 >	NOTE
0	0	0	0	ALL OFF	ALL OFF		TBD
0	0	0	1	X+ / X- off Y+ / Y- on	X+ / X- off Y+ / Y- on	1000	Y Position
0	0	1	0	ALL OFF	ALL OFF	0011	IN3
0	0	1	1	X+ / Y- off Y+ / X- on	X+ / Y- off Y+ / X- on	1000	Z_1 Position
0	1	0	0	X+ / Y- off Y+ / X- on	X+ / Y- off Y+ / X- on	1010	Z_2 Position
0	1	0	1	X+ / X- on Y+ / Y- off	X+ / X- on Y+ / Y- off	1001	X Position
0	1	1	0	ALL OFF	ALL OFF	0100	IN4
0	1	1	1	ALL OFF	ALL OFF		TBD
1	0	0	0	ALL OFF	ALL OFF		TBD
1	0	0	1	X+ / X- off Y+ / Y- on	ALL OFF	1000	Y Position
1	0	1	0	ALL OFF	ALL OFF	0011	IN3
1	0	1	1	X+ / Y- off Y+ / X- on	ALL OFF	1000	Z_1 Position
1	1	0	0	X+ / Y- off Y+ / X- on	ALL OFF	1010	Z_2 Position
1	1	0	1	X+ / X- on Y+ / Y- off	ALL OFF	1001	X Position
1	1	1	0	ALL OFF	ALL OFF	0100	IN4
1	1	1	1	ALL OFF	ALL OFF		TBD

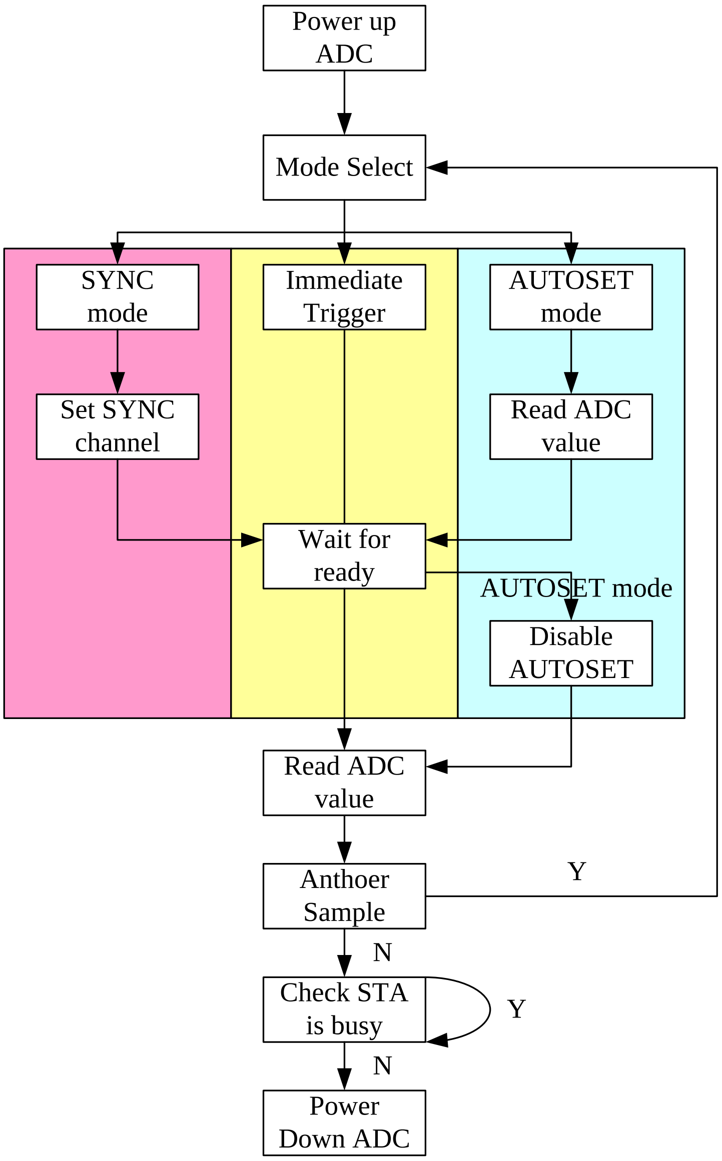
Table 28 Relationship between commands and touch panel control signals

Background Detection:

In order to monitor at AP sleep mode, Auxadc adds the background detection function to monitor the pre-defined channel. If the expected times is achieved, higher or lower than expected voltage, Auxadc will issue LOWBAT interrupt to AP interrupt controller to wake up the AP. Besides AP sleep mode, this background detection is not recommended to be used in the normal mode. In the normal mode, SW is recommended to regularly measure battery voltage. The LOWBAT interrupt ID refers to AP interrupt controller. Moreover, LOWBAT waking up AP needs to set enable bit in the sleep controller and please refer to AP sleep controller to enable it before entering the sleep mode.

Power up/down ADC procedure:

 © 2009 MediaTek Inc.


Power up ADC
Mode Select
SYNC mode
Immediate Trigger
AUTOSET mode
Set SYNC channel
Read ADC value
Wait for ready
AUTOSET mode
Disable AUTOSET
Read ADC value
Anthoer Sample
Y
N
Check STA is busy
Y
N
Power Down ADC

2.10.1 Register Definitions

Register Address	Register Function	Acronym
0x8003D000	Auxiliary ADC control register 0	AUXADC_CON0
0x8003D004	Auxiliary ADC control register 1	AUXADC_CON1
0x8003D008	Auxiliary ADC control register 2	AUXADC_CON2
0x8003D010	Auxiliary ADC channel 0 data register	AUXADC_DAT0
0x8003D014	Auxiliary ADC channel 1 data register	AUXADC_DAT1
0x8003D018	Auxiliary ADC channel 2 data register	AUXADC_DAT2
0x8003D01C	Auxiliary ADC channel 3 data register	AUXADC_DAT3
0x8003D020	Auxiliary ADC channel 4 data register	AUXADC_DAT4
0x8003D024	Auxiliary ADC channel 5 data register	AUXADC_DAT5
0x8003D028	Auxiliary ADC channel 6 data register	AUXADC_DAT6
0x8003D02C	Auxiliary ADC channel 7 data register	AUXADC_DAT7
0x8003D030	Auxiliary ADC channel 8 data register	AUXADC_DAT8
0x8003D034	Auxiliary ADC channel 0 data register for TDMA event 0	AUXADC_DAT9
0x8003D050	Touch Screen Debounce Time	AUX_TS_DEBT
0x8003D054	Touch Screen Sample Command	AUX_TS_CMD
0x8003D058	Touch Screen Control	AUX_TS_CON
0x8003D060	Auxadc Background Voltage threshold	AUX_DET_VOLT
0x8003D064	Auxadc Background Detected Channel	AUX_DET_SEL
0x8003D068	Auxadc Background Detection Period	AUXADC_DET_PERIOD
0x8003D06C	Auxadc Background Detection Debounce	AUX_DET_DEBT

Table 29 ***Auxadc Registers***

0x8003D000 Auxiliary ADC control register 0 AUXADC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SYN8	SYN7	SYN6	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

SYN*n* These 9 bits define whether the corresponding channel is sampled or not in timer-triggered mode. It is associated with timing offset register TDMA_AUXEV1. It supports multiple flags. The flags can be automatically cleared after those channel have been sampled if AUTOCLR1 in the register AUXADC_CON3 is set.

0 The channel is not selected.

1 The channel is selected.

 © 2009 MediaTek Inc.

0x8003D004 Auxiliary ADC control register 1 AUXADC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								IMM8	IMM7	IMM6	IMM5	IMM4	IMM3	IMM2	IMM1	IMM0
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

IMM*n* These 7 bits are set individually to sample the data for the corresponding channel. It supports multiple flags.

0 The channel is not selected.

1 The channel is selected.

0x8003D008 Auxiliary ADC control register 2 AUXADC_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SYN9
Type																R/W
Reset																0

SYN9 This bit is used only for channel 0 and is to be associated with timing offset register TDMA_AUXEV0 in the TDMA timer in timer-triggered mode. The flag can be automatically cleared after channel 0 has been sampled if AUTOCLR0 in the register AUXADC_CON3 is set.

0 The channel is not selected.

1 The channel is selected.

0x8003D00C Auxiliary ADC control register 3 AUXADC_CON3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTO SET				PUWAI T_EN		AUTO CLR1	AUTO CLR0								STA
Type	R/W				R/W		R/W	R/W								RO
Reset	0				0		0	0								0

AUTOSET This field defines the auto-sample mode of the module. In auto-sample mode, each channel with its sample register being read can start sampling immediately without configuring the control register AUXADC_CON1 again.

PUWAIT_EN Thus field enables the power warm-up period to ensure power stability before the SAR process takes place. It is recommended to activate this field.

0 The mode is not enabled.

1 The mode is enabled.

AUTOCLR1 The field defines the auto-clear mode of the module for event 1. In auto-clear mode, each timer-triggered channel gets samples of the specified channels once the SYN*n* bit in the register AUXADC_CON0 has been set. The SYN*n* bits are automatically cleared and the channel is not enabled again by the timer event except when the SYN*n* flags are set again.

0 The automatic clear mode is not enabled.

1 The automatic clear mode is enabled.

AUTOCLR0 The field defines the auto-clear mode of the module for event 0. In auto-clear mode, the timer-triggered channel 0 gets the sample once the SYN9 bit in the register AUXADC_CON2 has

been set. The SYN9 bit is automatically cleared and the channel is not enabled again by the timer event 0 except when the SYN9 flag is set again.

0 The automatic clear mode is not enabled.

1 The automatic clear mode is enabled.

STA The field defines the state of the module.

0 This module is idle.

1 This module is busy.

0x8003D010 Auxiliary ADC channel 0 register AUXADC_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT**									
Type							RO									
Reset							0									

The register stores the sampled data for the channel 0. There are 8 registers of the same type for the corresponding channel . The overall register definition is listed in **Table 30**.

Register Address	Register Function	Acronym
0x8003D010	Auxiliary ADC channel 0 data register	AUXADC_DAT0
0x8003D014	Auxiliary ADC channel 1 data register	AUXADC_DAT1
0x8003D018	Auxiliary ADC channel 2 data register	AUXADC_DAT2
0x8003D01C	Auxiliary ADC channel 3 data register	AUXADC_DAT3
0x8003D020	Auxiliary ADC channel 4 data register	AUXADC_DAT4
0x8003D024	Auxiliary ADC channel 5 data register	AUXADC_DAT5
0x8003D028	Auxiliary ADC channel 6 data register	AUXADC_DAT6
0x8003D02C	Auxiliary ADC channel 7 data register	AUXADC_DAT7
0x8003D030	Auxiliary ADC channel 8 data register	AUXADC_DAT8
0x8003D034	Auxiliary ADC channel 0 data register for TDMA event 0	AUXADC_DAT9

Table 30 ***Auxiliary ADC data register list***

0x8003D050 Touch Screen Debounce Time AUX_TS_DEBT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DEBOUNCE TIME**													
Type			R/W													
Reset			0													

DEBOUNCE TIME While the analog touch screen irq signal is from high to low level, auxadc will issue an interrupt after the debounce time.

0x8003D054 Touch Screen Sample Command AUX_TS_CMD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**ADDRESS**			**MODE**	**SE/DF**	**PD**	
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

ADDRESS Define which x or y or z data will be sampled.

- **001** Y Position
- **011** Z1 Position
- **100** Z2 Position
- **101** X Position
- **Others** Reserved

MODE Select the sample resolution

- **0** 10-bit resolution
- **1** 8-bit resolution

SE/DF Mode selection

- **0** Differential mode
- **1** Single-end mode

PD Power down control for analog IRQ signal and touch screen sample control signal

- **00** Turn on Y-_drive signal and PDN_sh_ref
- **01** Turn on PDN_IRQ and PDN_sh_ref
- **10** Reserved
- **11** Turn on PDN_IRQ

0x8003D058 Touch Screen Control AUX_TS_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**ST**	**SPL**
Type															R	R/W
Reset															0	0

SPL Touch Screen Sample Trigger

- **0** No Action
- **1** While SW writes 1'b1, auxadc will trigger the touch screen process. After the sample process of touch screen finishes, this bit will be disserted.

ST Touch Screen Status

- **0** Touch Screen is idle.
- **1** Touch Screen is touched.

0x8003D05C Touch Screen Sample DATA AUX_TS_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT**									
Type							RO									
Reset							0									

This register stores the touch screen sample data.

0X8003D060 Auxadc Background Voltage threshold AUX_DET_VOLT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INV**						**VOLT**									
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	0	0	0	0	0	0	0									

INV While the battery voltage is high or lower than pre-defined voltage (VOLT), the interrupt will be issued to AP.

- **0** Lower
- **1** Higher

VOLT Pre-defined voltage threshold

0X8003D064 Auxadc Background Detected Channel AUX_DET_SEL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													DET_CH			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0			

DET_CH Which a channel will be sampled in the background

0X8003D068 Auxadc Background Detection Period AUXADC_DET_PERIOD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DET_PERIOD													
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0													

DET_PERIOD Background sample period. While this value is not zero, the background detection will be activated automatically. (use 32k clock as unit)

0X8003D06C Auxadc Background Detection Debounce AUX_DET_DEBT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DEBOUNCE TIME													
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0													

DEBOUNCE_TIME While the number of that the detected channel is

2.11 CoreSight

2.11.1 General Description

CoreSight is one set of debug IP tool kit provided by ARM. It supports MCU & bus real-time trace, multi-core debug, and cross trigger feature. In MT6516 CoreSight system comprises of 1 DAP (debug access port), 1 ETM (embedded trace macrocell), 1 TPIU (trace port interface unit), 1 ETB (embedded trace buffer), 1 CTM (cross trigger matrix), and 3 CTI (cross trigger interface). ETM can be categorized as trace source, whereas TPIU and ETB are trace sink. Trace source records CPU activities and produces trace data. Through ATB (AMBA Trace Bus), trace sink collects these trace data to external debugger by TPIU or to internal storage (ETB). DAP is the gateway transmitting information between external debugger and dual core system through JTAG interface. DAP can control ARM7EJS, ARM926EJS and all CoreSight components through single JTAG port. ETM records real-time ARM926EJS operations, compresses, and transfers these trace data to ATB.

 © 2009 MediaTek Inc.

Replicator is a device which dispatches trace data to two different types of trace sink. TPIU and ETB both are used to store trace, while TPIU transfers to external debugger and ETB stores internally. CTM and CTI control cross triggering that transmit trigger event from one device to another.
The description above is not exhaustive. About the detailed information, programmer's model, and so on, please refer to CoreSight technical reference manual on ARM website.
Figure 17 shows an overview of the CoreSight system in MT6516.

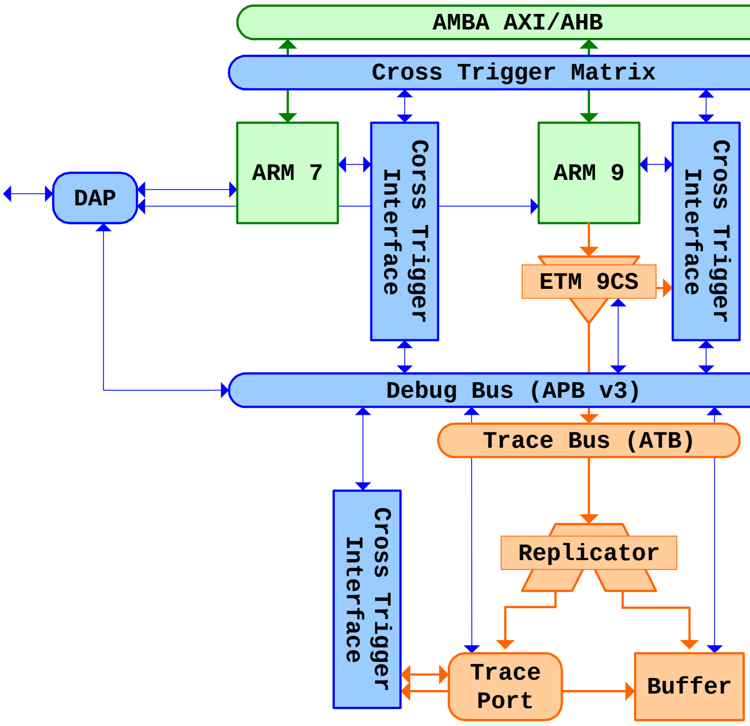


Figure 17 ***Overview of MT6516 CoreSight system***

2.11.2 Project-Dependent Specification

Some CoreSight specifications are project-dependent. The following features are only in MT6516.

- **TPIU trace port : trace data port + trace control port + trace clock port = 16 + 1 + 1 = 18 bits**
- **Trace clock : 52MHz**
- **ETB size : 4KB**
- **Cross trigger feature**

Figure 2 shows an overview of cross trigger in MT6516 CoreSight system.

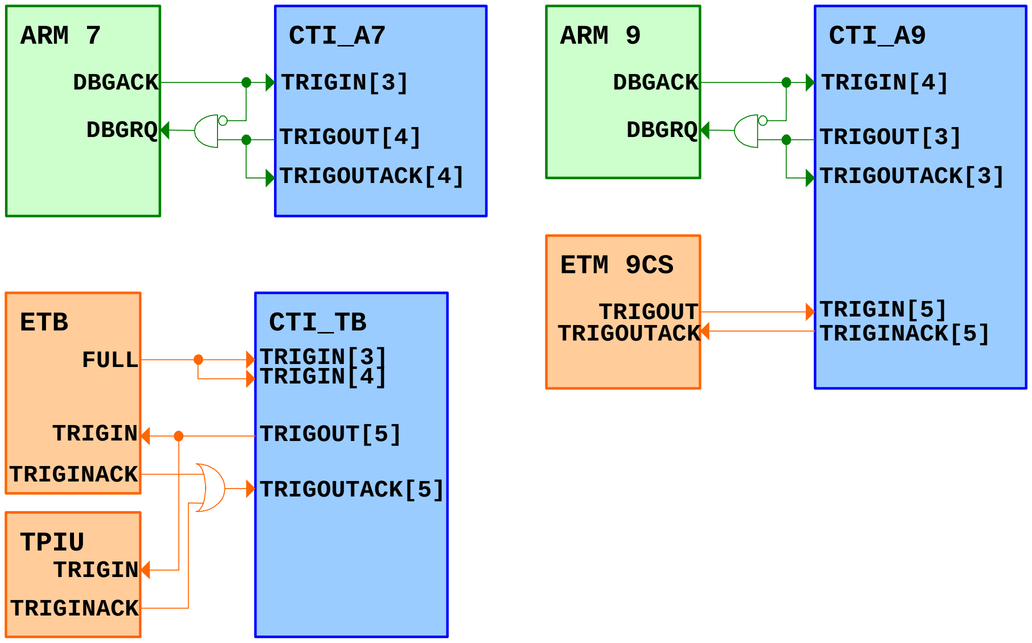


Figure 2 ***Cross Trigger Overview***

For example, if CTI_A7 enables TRIGIN[3] and CTI_A9 enables TRIGOUT[3], once ARM7 enters debug mode, DBGACK will be asserted. This assertion will be transferred through CTI_A7 → CTM → CTI_A9 → ARM9 DBGRQ signal and force ARM 9 also enters debug mode. This scenario achieves that one ARM core state influences the other one. There are several scenarios in MT6516 cross trigger function.

1. ARM7 & ARM9 enter debug mode in turn.
2. ETB full forces ARM7 & ARM9 to enter debug mode.
3. ETM_A9 issue trigger event and cause TPIU & ETB to stop tracing.

Cross trigger operational procedure in Trace32 (Take the above case for example)

Data.Set dap:8004302c %l 0x1 // Enable CTI_A7 TRIGIN[3] and assign to channel 0

Data.Set dap:80043000 %l 0x1 // Enable CTI_A7

Data.Set dap:800440ac %l 0x1 // Enable CTI_A9 TRIGOUT[3] and assign to channel 0

Data.Set dap:80044000 %l 0x1 // Enable CTI_A9

2.11.3 Register Definitions

Table 31 summarizes the base address of CoreSight components.

Base Address	CoreSight Component	Acronym
80040000h	Debug Access Port	**DAP**
80041000h	Embedded Trace Macrocell – ARM 9	**ETM9**
80042000h	Trace Port Interface Unit	**TPIU**
80043000h	Cross Trigger Interface – ARM 7	**CTIA7**
80044000h	Cross Trigger Interface – ARM 9	**CTIA9**

 © 2009 MediaTek Inc.

80045000h	Embedded Trace Buffer	**ETB**
80046000h	Cross Trigger Interface – TPIU & ETB	**CTITB**

Table 31 ***Base Address of CoreSight Components***

CoreSight technical reference manual has detailed register definition. Please download these documents from ARM's website : www.arm.com. Here only introduce the newly-added features.

TPIU+040h Trace Clock Delay Chain Control Register TPIU_DLYCHN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DELAY SELECT SETTING**															
Type	R/W															
Reset	1															

DELAY SELECT SETTING

Trace port has three types of signals : TRACECLK, TRACEDATA, TRACECTL, and need to connect to external debugger. TRACECLK needs flexibility to adjust timing to sample correct trace data. This setting decides how much delay TRACECLK pin will pass through. Only sets one bit at a time, such as 16'h1, 16'h2, 16'h4, etc. The larger this setting is, the more delay TRACECLK passes through. Default value is 16'h1, which passes one tap of delay cell.

TPIU+080h Trace Clock Positive Negative Control Register TPIU_TCKPN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**PN**
Type																R/W
Reset																1

PN TRACECLK is generated by one double-frequency clock : TRACECLKIN. This bit decides to sample at positive or negative edge to create TRACECLK. Default uses negative edge to sample.

2.11.4 Application Note

Clock of CoreSight system is controlled by power down bit. If using Trace32 to connect CoreSight (PAD icoresight = 1), CoreSight system will be powered up by default. If using software to control CoreSight (PAD icoresight = 0), clear CoreSight power down bit is required.

*APMCUSYS_PDN_CLR1 (0x800393a0) = 0x1

Also, because trace port 18 signals are muxed with GPIO, please configure GPIO mode register to switch to trace port first.

*GPIO_MODE13 (0x800026c0) = *GPIO_MODE13 & 0xf0 | 0xff0f

*GPIO_MODE14 (0x800026d0) = *GPIO_MODE14 & 0xfff0 | 0xf

*GPIO_MODE18 (0x80002710) = *GPIO_MODE18 & 0x3 | 0x5554

*GPIO_MODE19 (0x80002720) = *GPIO_MODE19 & 0xffc0 | 0x15

2.12 CPU-CPU interface (CCIF)

2.12.1 General Description and Features

CCIF is designed for communication between two CPUs. Message from any one CPU can be sent to another CPU efficiently. The functional block diagram of CCIF is shown in fig. 1. And the operation flow of CCIF is shown in fig. 2.

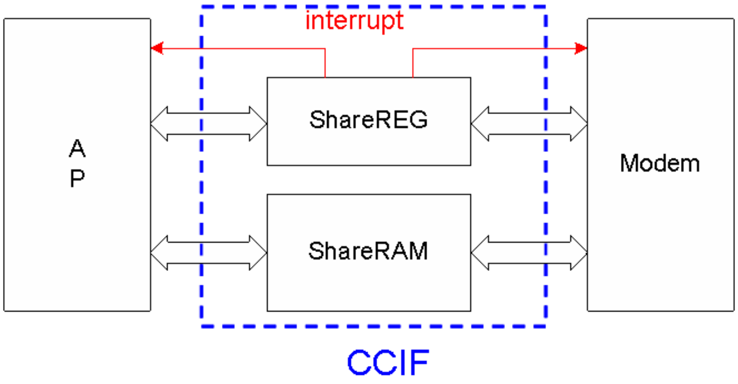


fig. 1

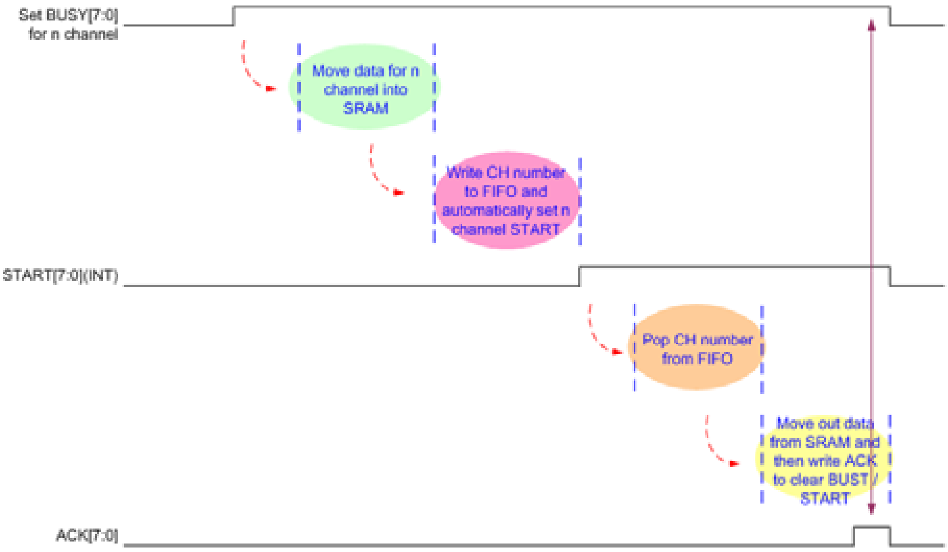


fig. 2

The flow of CCIF protocol can be summarized as:

- Transmission side: Set ***busy*** of channel #n
- Transmission side: Move data to channel #n
- Transmission side: Write channel number to FIFO (***Start*** of channel #n will be set automatically)

- Interrupt issued to Reception side automatically
- Reception side: Read channel number from FIFO
- Reception side: Move data and ***ACK***
- ***Busy*** and ***Start*** is cleared automatically

The main features of the CCIF are:

- Two CCIF sides can have different clock frequency
- Two different arbitration modes are supported (sequential mode or arbitration by MCU)
- One internal 256 byte dual port SRAM
- 16 channels are provided (8 channels AP→MD, 8 channels MD→AP)
- Support 1T/2T APB read/write

2.12.2 Register Definitions of APMCU side

Table 1 is the AP CPU-CPU interface (CCIF) register mapping table that summarizes the AP CCIF register address mapping on APB bus and function description. Note: The CEVA CCIF base address of the AP side is 0xB1001000, other offset registers of CEVA CCIF is same as the following.

APB Address	Register Function	Acronym
8003_6000h	AP CCIF control register	**CON**
8003_6004h	AP CCIF busy register	**BUSY**
8003_6008h	AP CCIF start register	**START**
8003_600Ch	AP CCIF transmit channel number register	**TCHNUM**
8003_6010h	AP CCIF receive channel number register	**RCHNUM**
8003_6014h	AP CCIF acknowledge register	**ACK**
8003_6100h -- 8003_61FFh	AP CCIF channel data register	**CHDATA**

Table 1 The AP CCIF register mapping

8003_6000h AP CCIF Control Register APCCIF_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ARB**
Type																R/W
Reset																0

ARB Enable arbitration mechanism for APMCU to decide which channel wants to be read.

 © 2009 MediaTek Inc.

0 Use sequential mechanism to decide which channel needs to be read firstly. RCHNUM represents the number of channel need to be processed by APMCU in advance. Only use three bits ([2:0]) in RCHNUM. The processing order of channels is only in time sequence.

RCHNUM[2:0]	Having data in which channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

1 APMCU decides to read which channel by arbitration. The bit value of RCHNUM ([7:0]) represents which channels need to be processed respectively.

RCHNUM[0]	Have data in channel 0?
0	No
1	Yes
RCHNUM[1]	**Have data in channel 1?**
0	No
1	Yes
RCHNUM[2]	**Have data in channel 2?**
0	No
1	Yes
RCHNUM[3]	**Have data in channel 3?**
0	No
1	Yes
RCHNUM[4]	**Have data in channel 4?**
0	No
1	Yes
RCHNUM[5]	**Have data in channel 5?**
0	No
1	Yes
RCHNUM[6]	**Have data in channel 6?**
0	No
1	Yes
RCHNUM[7]	**Have data in channel 7?**
0	No
1	Yes

© 2009 MediaTek Inc.

8003_6004h AP CCIF Busy Register APCCIF_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUSY							
Type									R/W							
Reset									0							

This register provides the status of channel 7-0 busy.

BUSY Indicate which channel operation is in process. It will de-assert when MDMCU finishes fetching channel data and then writes acknowledge for according channel. BUSY [7] represents the busy status of channel 7, and so on.

8003_6008h AP CCIF Start Register APCCIF_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									START							
Type									R							
Reset									0							

This register provides the status of channel 7-0 start. It is only for debugging usage.

START Indicate the state of finishing transmitting channel number but not receiving acknowledge. It will de-assert when writing acknowledge for according channel by MDMCU. START [0] represents the start status of channel 0, and so on.

8003_600Ch AP CCIF Transmit Channel Number Register APCCIF_TCHNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TCHNUM		
Type														W		
Reset														0		

This register specifies the transmitted channel number set by APMCU.

TCHNUM The 3-bit channel number represents which channel (channel 7-0) is to be used for transmitting data to MDMCU.

 © 2009 MediaTek Inc.

8003_6010h AP CCIF Receive Channel Number Register APCCIF_RCHNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**RCHNUM**		
Type														R		
Reset														0		

This register provides the received channel number read-out by APMCU.

RCHNUM The 3-bit channel number represents which channel (channel 7-0) is to be used for receiving data from MDMCU.

8003_6014h AP CCIF Acknowledge Register APCCIF_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**ACK**							
Type									W							
Reset									0							

This register specifies acknowledge when finishing one channel receiving.

ACK Acknowledge. It is write-cleared and set by APMCU for clearing interrupt status and BUSY/START states of MDMCU CPU-CPU interface according channel. Writing ACK is the latest step for ending one channel transfer.

8003_6100h AP CCIF Channel Data Register APCCIF_CHDATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DATA[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA[15:0]**															
Type	R/W															

This register specifies the data port for channel data accessing.

DATA The 32-bit channel data read from one 256 bytes dual-port SRAM. This SRAM can also be accessed by MDMCU subsystem. Total 256 bytes of channel data are read/written from address: APCCIF+0100h to address: APCCIF+01ffh by APMCU.

2.12.3 Register Definitions of MDMCU side

Table 1 is the MD CPU-CPU interface (CCIF) register mapping table that summarizes the MD CCIF register address mapping on APB bus and function description.

APB Address	Register Function	Acronym
8116_0000h	MD CCIF control register	CON
8116_0004h	MD CCIF busy register	BUSY
8116_0008h	MD CCIF start register	START
8116_000Ch	MD CCIF transmit channel number register	TCHNUM
8116_0010h	MD CCIF receive channel number register	RCHNUM
8116_0014h	MD CCIF acknowledge register	ACK
8116_0100h -- 8116_01FFh	MD CCIF channel data register	CHDATA

Table 1 The MD CCIF register mapping

8116_0000h MD CCIF Control Register MDCCIF_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name																ARB
Type																R/W
Reset																0

ARB Enable arbitration mechanism for MDMCU to decide which channel wants to be read.

0 Use sequential mechanism to decide which channel needs to be read firstly. RCHNUM represents the number of channel need to be processed by MDMCU in advance. Only use three bits ([2:0]) in RCHNUM. The processing order of channels is only in time sequence.

RCHNUM[2:0]	Having data in which channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

1 MDMCU decides to read which channel by arbitration. The bit value of RCHNUM ([7:0]) represents which channels need to be processed respectively.

RCHNUM[0]	Have data in channel 0?

0	No
1	Yes
RCHNUM[1]	**Have data in channel 1?**
0	No
1	Yes
RCHNUM[2]	**Have data in channel 2?**
0	No
1	Yes
RCHNUM[3]	**Have data in channel 3?**
0	No
1	Yes
RCHNUM[4]	**Have data in channel 4?**
0	No
1	Yes
RCHNUM[5]	**Have data in channel 5?**
0	No
1	Yes
RCHNUM[6]	**Have data in channel 6?**
0	No
1	Yes
RCHNUM[7]	**Have data in channel 7?**
0	No
1	Yes

8116_0004h MD CCIF Busy Register MDCCIF_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name									**BUSY**							
Type									R/W							
Reset									0							

This register provides the status of channel 7-0 busy.

BUSY Indicate which channel operation is in process. It will de-assert when APMCU finishes fetching channel data and then writes acknowledge for according channel. BUSY [7] represents the busy status of channel 7, and so on.

8116_0008h MD CCIF Start Register MDCCIF_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									START							
Type									R							
Reset									0							

This register provides the status of channel 7-0 start. It is only for debugging usage.

START Indicate the state of finishing transmitting channel number but not receiving acknowledge. It will de-assert when writing acknowledge for according channel by APMCU. START [0] represents the start status of channel 0, and so on.

8116_000Ch MD CCIF Transmit Channel Number Register MDCCIF_TCHNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TCHNUM		
Type														W		
Reset														0		

This register specifies the transmitted channel number set by MDMCU.

TCHNUM The 3-bit channel number represents which channel (channel 7-0) is to be used for transmitting data to APMCU.

8116_0010h MD CCIF Receive Channel Number Register MDCCIF_RCHNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RCHNUM		
Type														R		
Reset														0		

This register provides the received channel number read-out by MDMCU.

RCHNUM The 3-bit channel number represents which channel (channel 7-0) is to be used for receiving data from APMCU.

8116_0014h MD CCIF Acknowledge Register MDCCIF_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ACK							

Type									W							
Reset									0							

This register specifies acknowledge when finishing one channel receiving.

ACK Acknowledge. It is write-cleared and set by MDMCU for clearing interrupt status and BUSY/START states of APMCU CPU-CPU interface according channel. Writing ACK is the latest step for ending one channel transfer.

8116_0100h MD CCIF Channel Data Register MDCCIF_CHDATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DATA[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA[15:0]**															
Type	R/W															

This register specifies the data port for channel data accessing.

DATA The 32-bit channel data read from one 256 bytes dual-port SRAM. This SRAM can also be accessed by APMCU subsystem. Total 256 bytes of channel data are read/written from address: MDCCIF+0100h to address: MDCCIF+01ffh by MDMCU.

2.13 EFUSE Controller (efusec)

2.13.1 General Description

There are some EFUSE macros in the chip. EFUSE macro is a one-time-programming non-volatile memory. We usually use it as storage of sensitive and important data. EFUSE controller delivers EFUSE status and re-initializes EFUSE macro. You can program the EFUSE via EFUSE controller with proper configuration and sequences.

2.13.2 Register Definitions

EFUSEC+0000h EFUSE control EFUSEC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**ESEL**			**WSEL**						**RD**	**BUSY**	**VLD**
Type					R/W			R/W						WO	RO	RO
Reset					0			0						0	0	0

VLD Indicate if EFUSE data are valid or not. EFUSEC will initialize all EFUSE macros automatically. After finishing the initialization, this bit will change to 1 from 0. In other case, if you initialize EFUSE macros by write RD=1 manually, the VLD will go to low. After RD process done, VLD will go to high again.

BUSY EFUSE controller is busy. You can write EFUSEC control registers only when BUSY is low.

RD Initialize EFUSE macros manually. The BUSY is 1 and VLD is 0 while EFUSEC re-initialize all EFUSE macros. After finishing the initialization, BUSY changes to 0 and VLD changes to 1. Please perform read and write operations when PLL is off.

WSEL EFUSE word selection. There are 2 words in each EFUSE macro. You should decide which word you will program

ESEL EFUSE macro selection. There are 6 EFUSE macros in the system. You should decide which macro you will program

ESEL	WSEL	EFUSE_Dx
011	0	EFUSE_D0
011	1	EFUSE_D1
100	0	EFUSE_D2
100	1	EFUSE_D3
101	0	EFUSE_D4
101	1	EFUSE_D5
110	0	EFUSE_D6
110	1	EFUSE_D7
111	0	EFUSE_D8
111	1	EFUSE_D9

EFUSEC+0004h EFUSE write data EFUSEC_WDAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WDAT[31:16]															
Type	R/W															
Reset	N/A															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WDAT[15:0]															
Type	R/W															
Reset	N/A															

WDAT After setting the EFUSE_SEL and WSEL, you can write WDAT with values you want to program. Once you write EFUSEC_WDAT, EFUSEC starts blowing EFUSE operation. The BUSY flag rises. After the EFUSEC finished the blowing process, the BUSY flag lowers. You can follow the guidelines below:

1. Wait until BUSY is 0
2. Set VFSOURCE=2.8V from Ground
3. Set ESEL and WSEL.
4. Write EFUSEC_WDAT with your prefer value.
5. Wait until BUSY is 0
6. If you want to blow other EFUSE macro, jump step 1
7. Set VFSOURCE= Ground
8. Write RD = 1. Wait BUSY=0 and VALID=1. Check the EFUSE contents you blew.

Note1: each bit valued 1 in WDAT means a blowing operation. Blown bits can not be blown again. Such that the final EFUSE content should be the original EFUSE content OR WDAT.

Note2: Please perform read and write operations when PLL is off.

EFUSEC+0008h EFUSE programming time EFUSEC_PGMT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		**PGMT[14:0]**														
Type		WO														
Reset		0x82														

PGMT This value defines the programming time.

EFUSEC+0010h EFUSE data out EFUSE_D0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EFUSE_D0**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**EFUSE_D0**															
Type	R/W															
Reset	0															

EFUSEC+0014h EFUSE data out EFUSE_D1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EFUSE_D1**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**EFUSE_D1**															
Type	R/W															
Reset	0															

EFUSEC+0018h EFUSE data out EFUSE_D2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EFUSE_D2**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**EFUSE_D2**															
Type	R/W															
Reset	0															

EFUSEC+001ch EFUSE data out EFUSE_D3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D3															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D3															
Type	R/W															
Reset	0															

EFUSEC+0020h EFUSE data out EFUSE_D4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D4															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D4															
Type	R/W															
Reset	0															

EFUSEC+0024h EFUSE data out EFUSE_D5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D5															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D5															
Type	R/W															
Reset	0															

EFUSEC+0028h EFUSE data out EFUSE_D6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D6															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D6															
Type	R/W															
Reset	0															

© 2009 MediaTek Inc.

EFUSEC+002ch EFUSE data out EFUSE_D7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D7															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D7															
Type	R/W															
Reset	0															

EFUSEC+0030h EFUSE data out EFUSE_D8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EFUSE_D8															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D8															
Type	R/W															
Reset	0															

EFUSEC+0034h EFUSE data out EFUSE_D9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	D89WP	D67WP	D45WP	EFUSE_D9												
Type	R/W	R/W	R/W	R/W												
Reset	0	0	0	0												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EFUSE_D9															
Type	R/W															
Reset	0															

EFUSE_D_X EFUSE output data

D_{XY}WP Write protection bit for EFUSE_D_X and EFUSE_D_Y

- **0** EFUSE_D_X and EFUSE_D_Y can be blown.
- **1** EFUSE_D_X and EFUSE_D_Y can not be blown.

2.14 External Memory Interface

2.14.1 General Description

MT6516 incorporates a powerful and flexible memory controller, External Memory Interface, to connect with a variety of memory components. This controller provides one generic access scheme for FLASH Memory, SRAM, PSRAM and. Up to 4 memory banks can be supported simultaneously, BANK0-BANK3, with a maximum size of 128MB each. This controller also provides another access scheme for DRAM (SDR/DDR), and 4 banks can be supported for simultaneous access, with a maximum size of 256MB of each bank.

The software program can treat different components by simply specifying certain predefined parameters. All these parameters are based on cycle time of system clock.

The interface definition based on such scheme is listed in **Table 32**. Note that, this interface always operates data in Little Endian format for all types of accesses.

Signal Name	Type	Description
XADMUX	I	Define ADMUX or not in NOR flash / PSRAM
EWAIT	I	Wait Signal Input
ED[31:0]	I/O	Data Bus
EA[26:0]	I/O	Address Bus
EDQS[3:0]	I/O	Data strobe in DDR
ECS# [3:0]	O	BANK3~BANK0 Selection Signal
EWR#	O	Write Enable Strobe
ERD#	O	Read Enable Strobe
EDQM[3:0]#	O	Data mask
EADV#	O	Burst Mode FLASH Memory Address Latch Signal
ERAS#	O	Row address latch signal (SDR/DDR)
ECAS#	O	Column address latch signal (SDR/DDR)
ECKE#	O	DRAM clock enable signal (SDR/DDR)
EC_CLK	O	Burst Mode FLASH/PSRAM Memory Clock Signal
ED_CLK	O	DRAM clock signal
ED_CLK_B	O	DRAM clock invert signal (for DDR)

***Table 32** External Memory Interface of MT6516*

REGISTER ADDRESS	REGISTER NAME	SYNONYM
EMI + 0000h	PSRAM controller register for BANK0	EMI_CONA
EMI + 0008h	PSRAM controller register for BANK1	EMI_CONB
EMI + 0010h	PSRAM controller register for BANK2	EMI_CONC
EMI + 0018h	PSRAM controller register for BANK3	EMI_COND
EMI + 0020h	PSRAM controller register for BANK0	EMI_CONE
EMI + 0028h	PSRAM controller register for BANK1	EMI_CONF
EMI + 0030h	PSRAM controller register for BANK2	EMI_CONG
EMI + 0038h	PSRAM controller register for BANK3	EMI_CONH
EMI + 0040h	DRAM MR/EMR	EMI_CONI
EMI + 0048h	DRAM controller timing configuration I	EMI_CONJ
EMI + 0050h	DRAM controller timing configuration II	EMI_CONK
EMI + 0058h	DRAM controller read data path configuration	EMI_CONL
EMI + 0060h	EMI Digital DLL Control	EMI_CONM

© 2009 MediaTek Inc.

EMI + 0068h	DRAM controller function configuration	EMI_CONN
EMI + 0070h	EMI General Control Register A	EMI_GENA
EMI + 0078h	EMI General Control Register B	EMI_GENB
EMI + 0080h	EMI General Control Register C	EMI_GENC
EMI + 0088h	EMI General Control Register D	EMI_GEND
EMI + 0090h	Modem side offset address	EMI_GENE
EMI + 0098h	EMI In/Out Delay Line Control	EMI_DELA
EMI + 00A0h	EMI In/Out Delay Line Control	EMI_DELB
EMI + 00A8h	EMI In/Out Delay Line Control	EMI_DELC
EMI + 00B0h	EMI In/Out Delay Line Control	EMI_DELD
EMI + 00B8h	EMI In/Out Delay Line Control	EMI_DELE
EMI + 00C0h	EMI In/Out Delay Line Control	EMI_DELF
EMI + 00C8h	EMI In/Out Delay Line Control	EMI_DELG
EMI + 00D0h	EMI In/Out Delay Line Control	EMI_DELH
EMI + 00D8h	EMI In/Out Delay Line Control	EMI_DELI
EMI + 00E0h	EMI In/Out Delay Line Control	EMI_DELJ
EMI + 00E8h	MPU address set (region_0)	EMI_MPUA
EMI + 00F0h	MPU address set (region_1)	EMI_MPUB
EMI + 00F8h	MPU address set (region_2)	EMI_MPUC
EMI + 0100h	MPU address set (region_3)	EMI_MPUD
EMI + 0108h	MPU address set (region_4)	EMI_MPUE
EMI + 0110h	MPU address set (region_5)	EMI_MPUF
EMI + 0118h	MPU address set (region_6)	EMI_MPUG
EMI + 0120h	MPU address set (region_7)	EMI_MPUH
EMI + 0128h	MPU region access configuration	EMI_MPUI
EMI + 0130h	MPU region access configuration	EMI_MPUJ
EMI + 0138h	MPU status	EMI_MPUK
EMI + 0140h	MPU status	EMI_MPUL
EMI + 0148h	**MPU error address**	EMI_MPUM
EMI + 0150h	MPU Error AP IRQ	EMI_MPUN
EMI + 0158h	MPU Error MD IRQ	EMI_MPUO
EMI + 0160h	EMI bus monitor enable	EMI_BMEN
EMI + 0168h	EMI bus cycle counter	EMI_BCNT
EMI + 0170h	EMI transaction counter for all masters	EMI_TACT
EMI + 0178h	EMI transaction counter for selected masters	EMI_TSCT
EMI + 0180h	EMI double word counter for all masters	EMI_WACT
EMI + 0188h	EMI double word counter for selected masters	EMI_WSCT

EMI + 0190h	EMI bus-busy counter for all masters	EMI_BACT
EMI + 0198h	EMI bus-busy counter for selected masters	EMI_BSCT
EMI + 01A0h	EMI dummy read controls	EMI_DRCT
EMI + 01B0h	DQSI auto-tracking control for CS[0]	EMI_DQSA
EMI + 01B8h	DQSI auto-tracking control for CS[1]	EMI_DQSB
EMI + 01C0h	DQSI auto-tracking control for CS[2]	EMI_DQSC
EMI + 01C8h	DQSI auto-tracking control for CS[3]	EMI_DQSD
EMI + 01D0h	DQSI auto-tracking calibrating delay value	EMI_DQSV
EMI + 01E0h	Modem side control register	EMI_MDCL

Table 33 ***External Memory Interface Register Map***

 © 2009 MediaTek Inc.

2.14.2 Registers

+0000h~0018h Asynchronous PSRAM and flash control registers EMI_CONA~D

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PSIZE		ADMUX	ASRD	ASWR	APRD	AS_ADV			AS_SET			AP_SET		AS_CS_END	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DE_reset	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1
AD_reset	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AP_WAIT_1ST				AR_WAIT				AW_WAIT					AS_HOLD		ADVEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DE_reset	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0
AD_reset	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

+00020h ~0038h Synchronous/Asynchronous PSRAM and flash control registers EMI_CONE~H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RD_WAIT_EN	WR_WAIT_EN	SY_RD	SY_WR	SY_SET			SY_RD_WAIT			SY_HOLD			SY_CS_END		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WPLO	WRPS	LSS	SRAM_A_LAT /SRAM_RDPTR_TOG	RD_DEL_SEL		RD_XWAIT_N_EN	WR_XWAIT_N_EN	RD_XWAIT_P_EN	WR_XWAIT_P_EN	SY_WR_WAIT				ADMUX_2T_DV	DEMUX_2T_DV
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PSIZE Page size for AP_RD (page read mode), SY_RD (synchronous read mode) and SY_WR (synchronous write mode)

PSIZE	00	01	10
	8 byte	16 byte	32 byte

ADMUX ADMUX/AD-DeMUX memory selection

- **0** Non ADMUX type memory
- **1** ADMUX type memory

AS_RD Enable asynchronous read (Only one of AS_RD, AP_RD and SY_RD can be set to 1, others must be 0)

- **0** Turn off asynchronous read
- **1** Turn on asynchronous read

AS_WR Enable asynchronous write (Only one of AS_WR, and SY_WR can be set to 1, others must be 0)

- **0** Turn off asynchronous write
- **1** Turn on asynchronous write

AP_RD Enable asynchronous page read (Only one of AS_RD, AP_RD and SY_RD can be set to 1, others must be 0)

- **0** Turn off asynchronous page read (burst-page read)
- **1** Turn on asynchronous page read (burst-page read)

AS_ADV Adjust ADV time in every transaction of asynchronous mode. AS_ADV is only adjustable at AS_WR/AS_RD mode and ADVEN = 1. (Unit: cycle, default 0: 0T, 1:1T ……) ---- Fig-1

AS_SET Adjust init set up time in every transaction of asynchronous read/write mode. (Unit: cycle, the set value must > 1, 1:2T, 2:3T……) ---- Fig-2

AP_SET Adjust init set up time in every transaction of asynchronous page read mode. (Unit: cycle, the set value must > 1, 1:2T, 2:3T……) ---- Fig-4

AS_CS_END Adjust CS disable time in the end of every transaction in asynchronous mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-2

AP_WAIT_1st Adjust first wait time in every transaction of page read mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-4

AR_WAIT Adjust wait time in every transaction of asynchronous read mode. (Unit: cycle, the set value must > 1, 1:2T, 2:3T……) ---- Fig-3

AW_WAIT Adjust wait time in every transaction of asynchronous wait mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-2

AS_HOLD Adjust hold time in every transaction of asynchronous mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-2

ADVEN ADV control in asynchronous read / write

- **0** Disable
- **1** Enable

ADV_OUT wave form timing

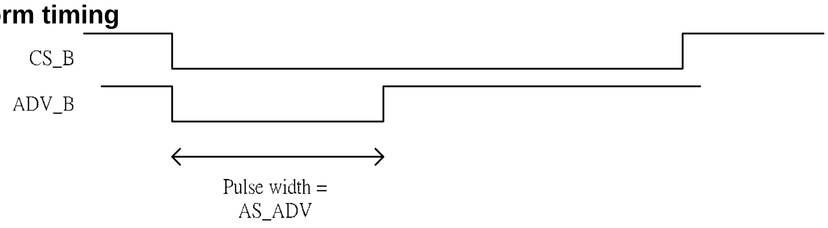
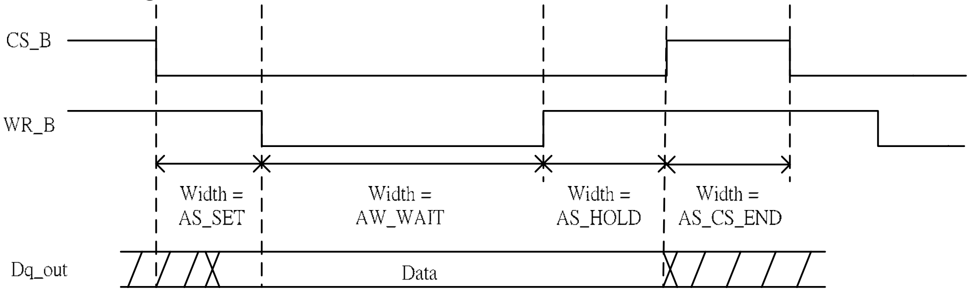


Fig-1

AS_WR wave form timing



 © 2009 MediaTek Inc.

Fig-2

AS_RD wave form timing

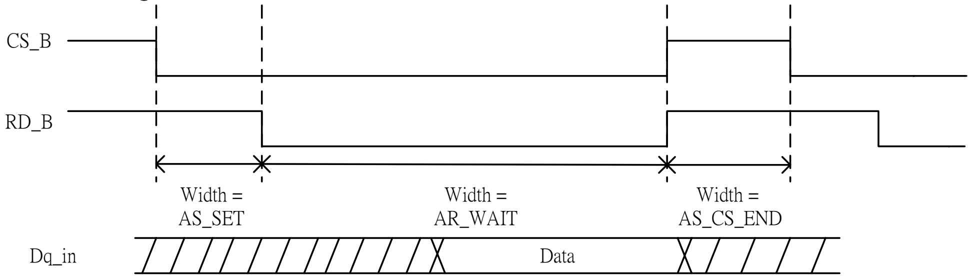


Fig-3

AP_RD wave form timing

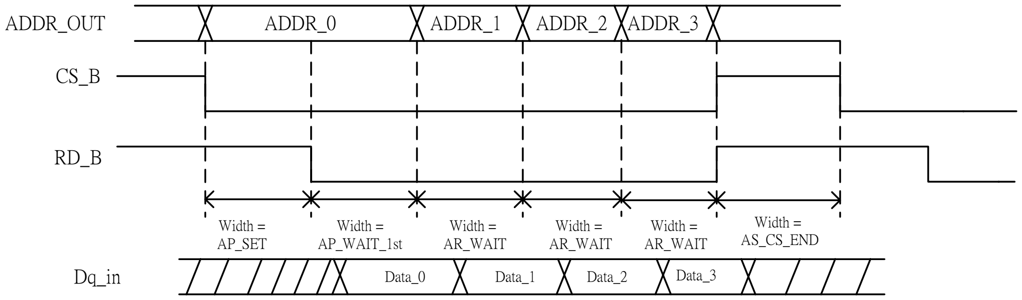


Fig-4

RD_WAIT_EN Monitor XWAIT signal from external memory to EMI controller at read

- **0** Not monitor
- **1** Monitor

WR_WAIT_EN Monitor XWAIT signal from external memory to EMI controller at write

- **0** Not monitor
- **1** Monitor

SY_RD Enable synchronous (burst) read (Only one of AS_RD, AP_RD and SY_RD can be set to 1, others must be 0)

- **0** Disable
- **1** Enable

SY_WR Enable synchronous (burst) write (Only one of AS_WR, and SY_WR can be set to 1, others must be 0)

- **0** Disable
- **1** Enable

SY_SET Adjust init set up time in every transaction of synchronous mode. (Unit: cycle, the set value must > 1, 1:2T, 2:3T…) ---- Fig-5

SY_RD_WAIT Adjust wait time in every transaction of synchronous read mode, after wait time passed the controller start to check input XWAIT signal (skip XWAIT unstable at the beginning). (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-5

SY_HOLD Adjust hold time in every transaction of synchronous mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-6

SY_CS_END Adjust CS disable time in the end of every transaction in synchronous mode. (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-5

SY_WR_WAIT Adjust wait time in every transaction of synchronous write mode, after wait time passed the controller start to check input XWAIT signal (skip XWAIT unstable at the beginning). (Unit: cycle, default 0: 1T, 1:2T ……) ---- Fig-6

SY_RD wave form timing

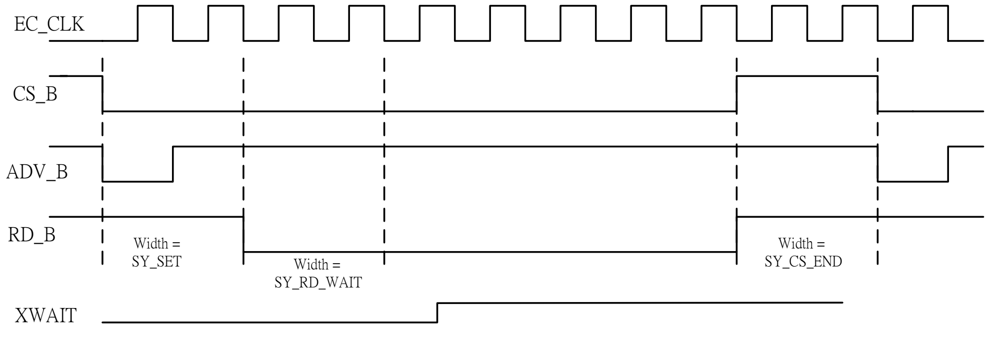


Fig-5

SY_WR wave form timing

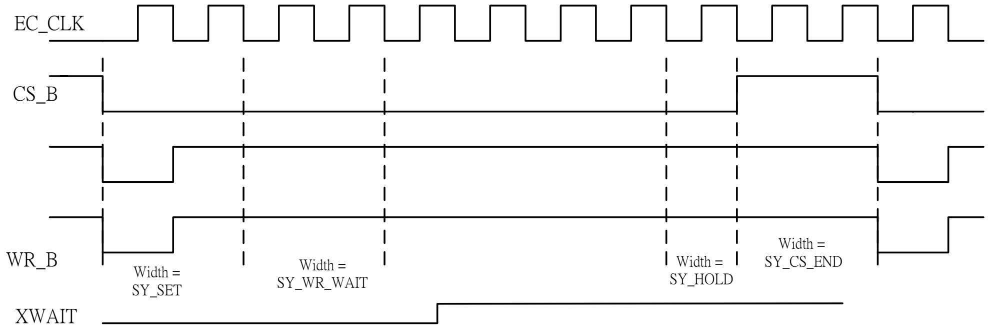


Fig-6

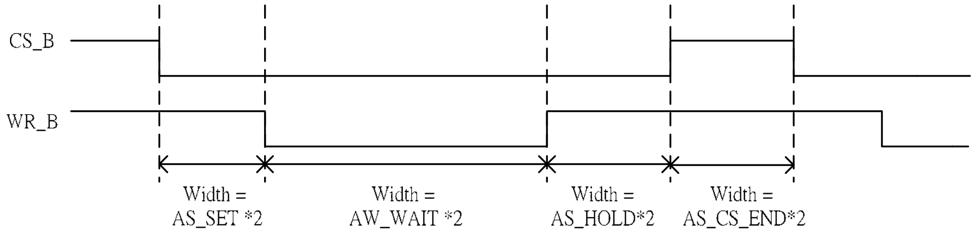
WPOL Enable XWAIT polarity change

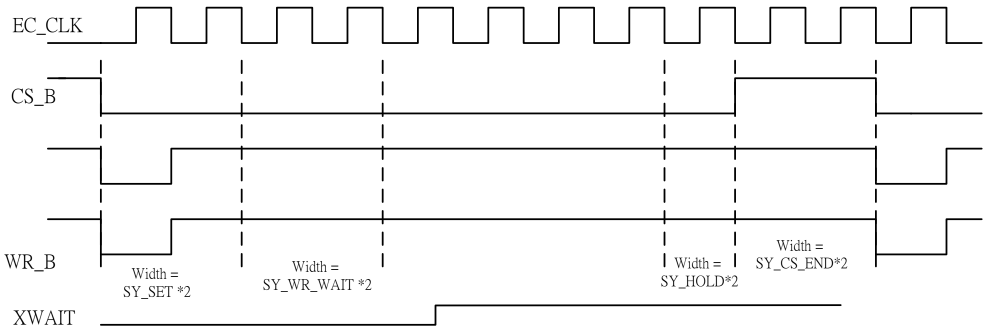
0 Disable

1 Enable

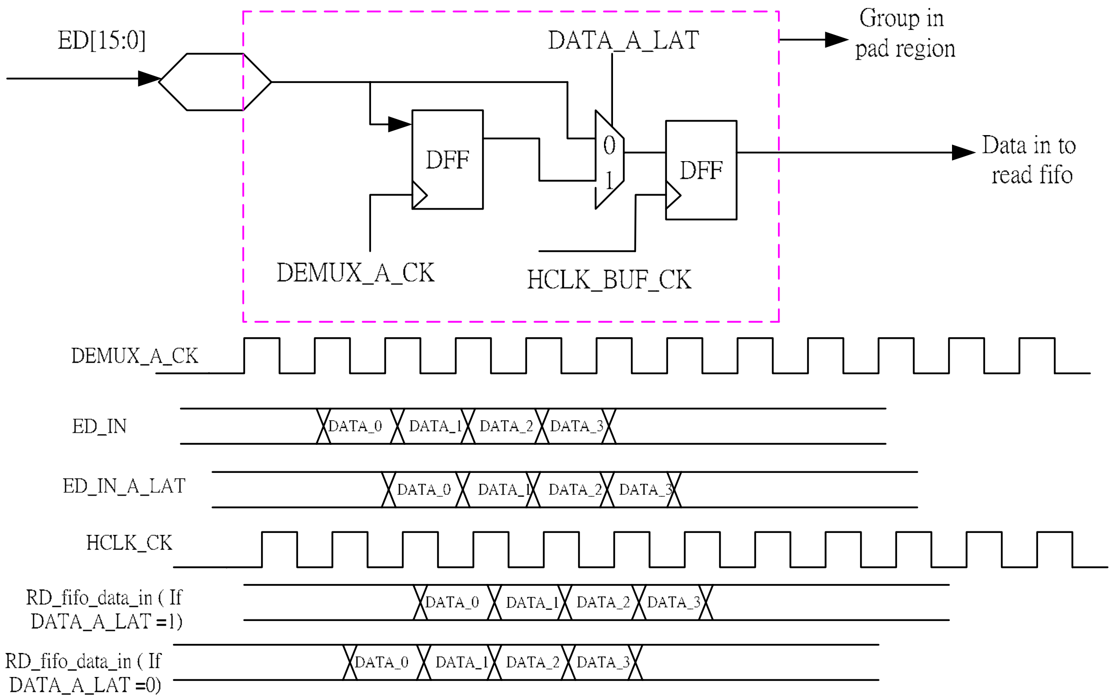
WRPS Access PSRAM by wrap mode. (Must be set to 1)

LSS Access PSRAM by half speed mode. If this bit is turned on, the output wave form timing will be doubled.





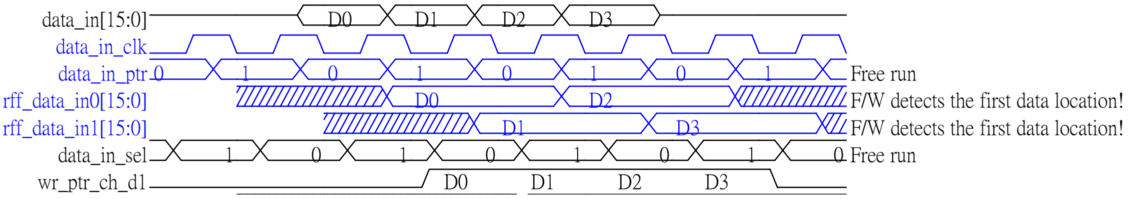
SRAM_A_LAT / SRAM_RDPTR_TOG SRAM_A_LAT: Sampling SRAM input data by internal adjustable clock first in synchronous read mode. (1T input data valid window when DEMUX_2T_DV = 0 or ADMUX_2T_DV = 0)



SRAM_RD_PTR_TOG: SRAM input data select (2T input data valid window when SRAM_2T_DV = 1). Refer to the "data_in_sel" signal as the following figure.

0 Do not toggle (default)

1 Toggle

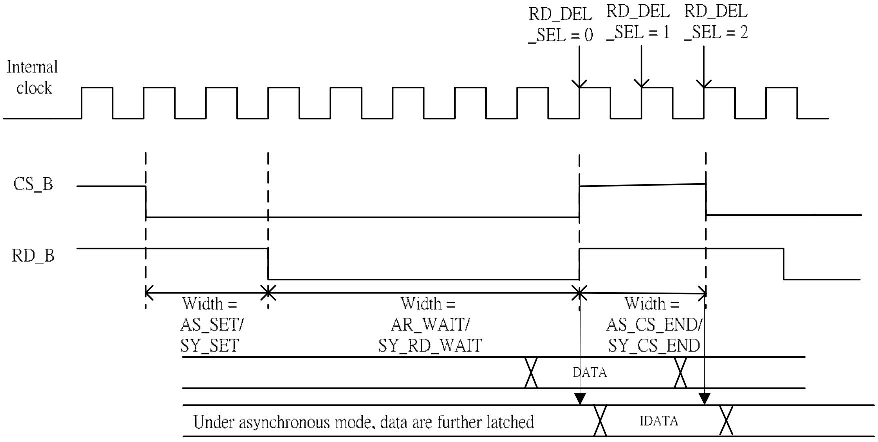


RD_DEL_SEL The input data latency to internal read FIFO

 © 2009 MediaTek Inc.

00 No delay
01 Delay 1T latch
10 Delay 2T latch
11 Reserved

Note that under the asynchronous/page read mode, RD_DEL_SEL must be set to 'h01.

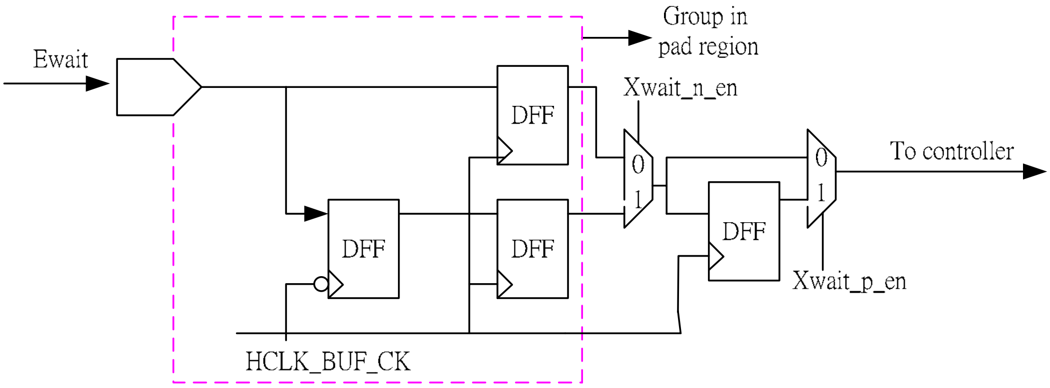


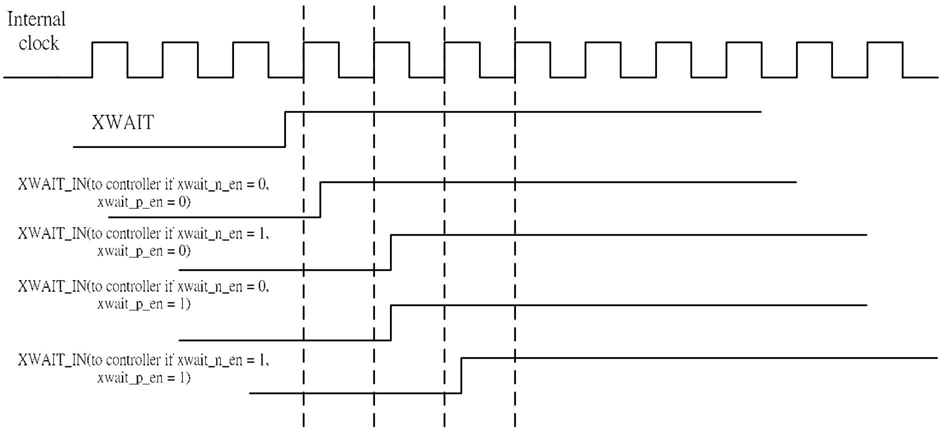
RD_XWAIT_N_EN Sampling input XWAIT signal by bus clock negative edge first in synchronous read mode.

WR_XWAIT_N_EN Sampling input XWAIT signal by bus clock negative edge first in synchronous write mode.

RD_XWAIT_P_EN Sampling input XWAIT signal by bus clock positive edge synchronous read mode.

WR_XWAIT_P_EN Sampling input XWAIT signal by bus clock positive edge synchronous write mode.





(Read/Write mode use separate XWAIT controller circuit)

SY_WR_WAIT Adjust wait time in every transaction of synchronous write mode. (0: 1clk, 1: 2clk ……)

ADMUX_2T_DV Input data valid window of ADMUX PSRAM

- **0** 1T (Default)
- **1** 2T (Suggest to enable at high-speed synchronous read, and PSRAM is set to the fixed latency mode)

DEMUX_2T_DV Input data valid window of AD-DeMUX PSRAM

- **0** 1T (Default)
- **1** 2T (Suggest to enable at high-speed synchronous read, and PSRAM is set to the fixed latency mode)

+0040h DRAM mode register set control registers EMI_CONI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**MBA1**	**MBA0**	**MA12**	**MA11**	**MA10**	**MA9**	**MA8**	**MA7**	**MA6**	**MA5**	**MA4**	**MA3**	**MA2**	**MA1**	**MA0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**EBA1**	**EBA0**	**EBA12**	**EBA11**	**EBA10**	**EBA9**	**EBA8**	**EBA7**	**EBA6**	**EBA5**	**EBA4**	**EBA3**	**EBA2**	**EBA1**	**EBA0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MBA1~0 DRAM bank address setting when load mode register to DRAM

MA12~0 DRAM mode register value

EBA1~0 DRAM bank address setting when load extended mode register to DRAM

EA12~0 DRAM extended mode register value

+0048h DRAM AC timing control 1 registers EMI_CONJ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**PRAL_CYC**			**REF_CYC**				**EXIT_SREF_CYC**						**LDMR_CYC**	
Type		R/W			R/W				R/W						R/W	
Reset		0			0				0						0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**ACT_RC_CYC**			**TRC_CYC**				**RTW_CYC**		**WR_WAIT_CYC**				**RD_WAIT_CYC**	
Type		R/W			R/W				R/W		R/W				R/W	

Reset		0	0	0	0			0

PRAL_CYC DRAM pre-charge cycle time (TRP) (Unit: cycle, default 0: 2T, 1:3T …, relative wave form please reference to DRAM spec)

REF_CYC DRAM refresh cycle time (TRFC) (Unit: cycle, default 0: 2T, 1:3T…, relative wave form please reference to DRAM spec)

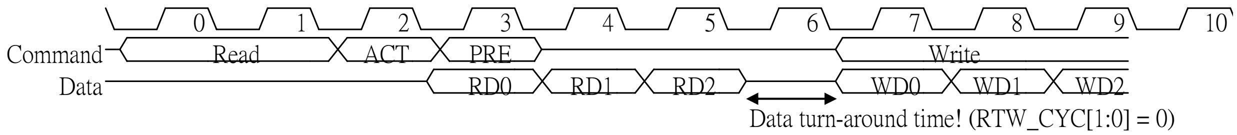
EXIT_SREF_CYC DRAM exit self refresh to first valid command cycle time (TXSR) (Unit: cycle, default 0: 2T, 1:3T ..., relative wave form please reference to DRAM spec)

LDMR_CYC DRAM load mode/extended-mode register cycle time (TMRD) (Unit: cycle, default 0: 1T, 1:2T …, relative wave form please reference to DRAM spec)

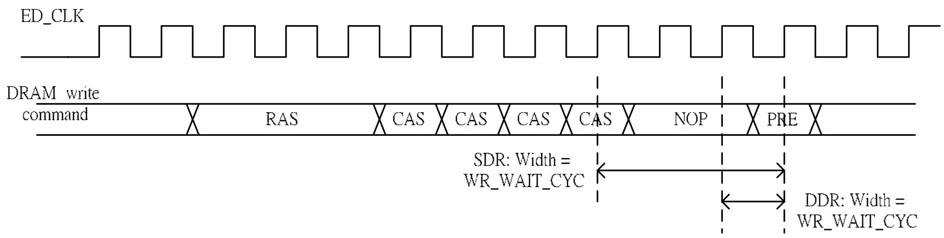
ACT_RC_CYC DRAM active to read/write command delay cycle time (TRCD) (Unit: cycle, the set value must >= 2, 2:3T, 3:4T…, relative wave form please reference to DRAM spec)

TRC_CYC DRAM active to active command (same bank) delay cycle time (TRC) (Unit: cycle, the set value must 0:2T , 2:4T, 3:5T…, relative wave form please reference to DRAM spec)

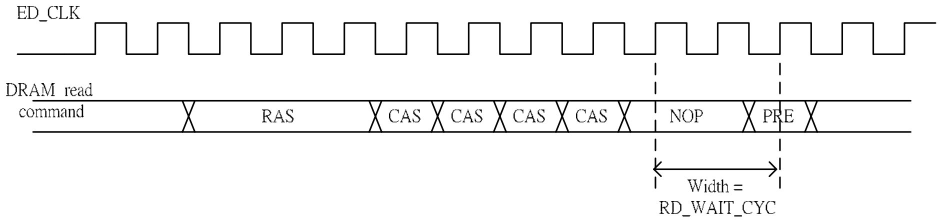
RTW_CYC Read to write data turn-around time (under inter-bank access) for preventing from contention of DRAM data bus. (Unit: cycle, 0/1/2/3 for delay 0/1/2/3T)



WR_WAIT_CYC DRAM write recovery cycle time (TWR). (Unit: cycle, default 0: 2T/0T, 1:3T/1T … for SDR/DDR, relative wave form please reference to SDR/DDR DRAM spec)



RD_WAIT_CYC DRAM read command to pre-charge delay cycle time (adjust the final read command to pre-charge command delay cycle time) (Unit: cycle, default 0: 1T, 1:2T …)



+0050h DRAM AC timing control 2 registers EMI_CONK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			REFP_CYC													
Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									SDR_RD0_PSEL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

REFP_CYC Auto refresh period cycle time (TREF). Note that REFP_CYC value must be larger than REF_CYC value (EMI_CONJ[27:24]).

Unit: cycle:

Default 0: 1T, 1:2T … when REF_CNT_EN (EMI_CONN[1]) = 1 & REF_FIX_CK (EMI_CONN[2]) = 0

Default 0: 0T, 1:1T … when REF_FIX_CK (EMI_CONN[2]) = 1,

relative waveform please reference to DRAM spec.

For example:

If one external DRAM having 8192 rows needs to do refresh at 64ms, thus the average refresh period is 64ms/8192 = 7.8us

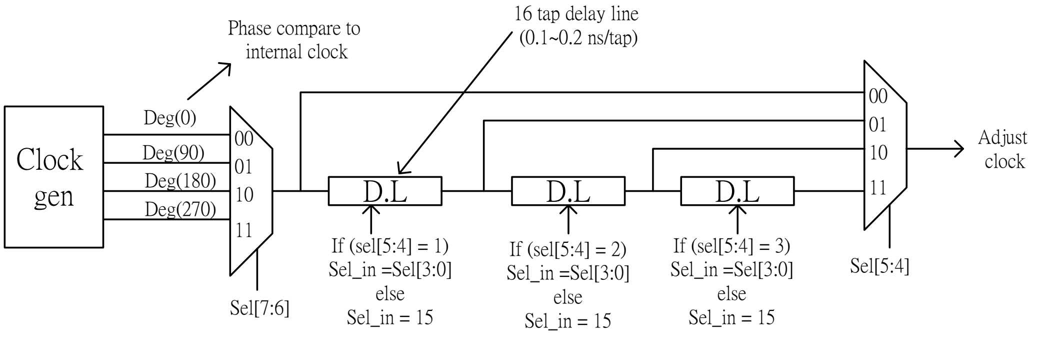
➔ If (REF_CNT_EN = 1 & REF_FIX_CK = 0): (REFP_CYC +1)* (clock period) must < 7.8us

➔ If REF_FIX_CK = 1: (REFP_CYC)* (1/3.25MHz) must < 7.8us,

to satisfy the DRAM refresh spec

SDR_RD0_PSEL Read phase delay for SDR input data bit [31:0] --- > 1 tape (0.1~0.2 ns)

SDR_RDx_PSEL is not adjustable at SDR_A_LAT = 0 and SDR_2T_DV = 0.

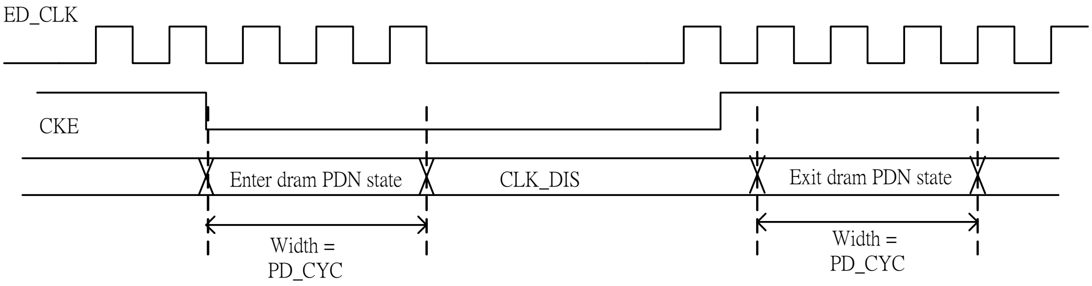


+0058h DRAM AC timing control 3 registers EMI_CONL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														PD_CYC		
Type														R/W		
Reset														0		
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		RAS_MIN_CYC							SDR_2T_DV	SDR_A_LAT / SDR_RDPTR_TOG				RD_DEL_SEL		
Type		R/W							R/W	R/W				R/W		
Reset		0					0		0	0				0		

PD_CYC Enter and exit DRAM power down state cycle time (Enter: 0/1/2…/7 = 1T/2T/3T…/8T, exit: 0/1/2…/7 = 1T/1T/2T…/7T)

 © 2009 MediaTek Inc.



RAS_MIN_CYC Active to pre-charge minimum cycle time (Unit: cycle, default 0: 2T, 1:3T …, relative wave form please reference to DRAM spec)

SDR_2T_DV Input data valid window of SDR PSRAM

- **0** 1T (Default)
- **1** 2T (Suggest to enable at high-speed read)

SDR_A_LAT / SDR_RDPTR_TOG SDR_A_LAT: Sampling SDR SRAM input data by internal adjustable clock first in synchronous read mode. (1T input data valid window when SDR_2T_DV = 0)

SDR DRAM input data (2T valid window, SDR_2T_DV = 1) selection

- **0** Do not toggle (default)
- **1** Toggle

RD_DEL_SEL The delay time cycles from read command for input read data to be sampled at RD_FIFO. (Include CAS latency, IO pad delay, PCB delay) (Unit: cycle, default 0: 1T, 1:2T …, 7:8T)

+0060h Digital DLL offset registers EMI_CONM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIGITAL_DLL_OFFSET_3				DIGITAL_DLL_OFFSET_2				DIGITAL_DLL_OFFSET_1				DIGITAL_DLL_OFFSET_0			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													DIGITAL_DLL_CAL_VALUE			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIGITAL_DLL_OFFSET_3 Digital DLL offset value for adding an offset delay on dqs_3 1/5 T delay path

DIGITAL_DLL_OFFSET_2 Digital DLL offset value for adding an offset delay on dqs_2 1/5 T delay path

DIGITAL_DLL_OFFSET_1 Digital DLL offset value for adding an offset delay on dqs_1 1/5 T delay path

DIGITAL_DLL_OFFSET_0 Digital DLL offset value for adding an offset delay on dqs_0 1/5 T delay path

DIGITAL_DLL_CAL_VALUE Digital DLL locking value for 1/5 T delay

+0068h DRAM mode register set and refresh control registers EMI_CONN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				PRAL_EN	AREF1_EN	AREF2_EN	LDMR_EN	LDEMR_EN	ADDR_SWAP	ADDR_TYPE					DRAM_TYPE	
Type				R/W	R/W	R/W	R/W	R/W	R/W	R/W					R/W	
Reset				0	0	0	0	0	0	0					0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	DQSI_CAL_MON				CAL_DONE	CAL_EN	SREF_ST	PDN_ST	SREF_EN	PDN_EN		REF_FIX_CK	REF_CNT_EN	DRAM_EN
Type	R/W				R	R/W	R	R	R/w	R/W		R/W	R/W	R/W
Reset	0				0	0	0	0	0	0		0	0	0

Five dram initialize steps must be executed step by step :

PRAL_EN Single pre-charge all enable (for DRAM initialize)

ARF1_EN Single auto-refresh-1 enable (for DRAM initialize)

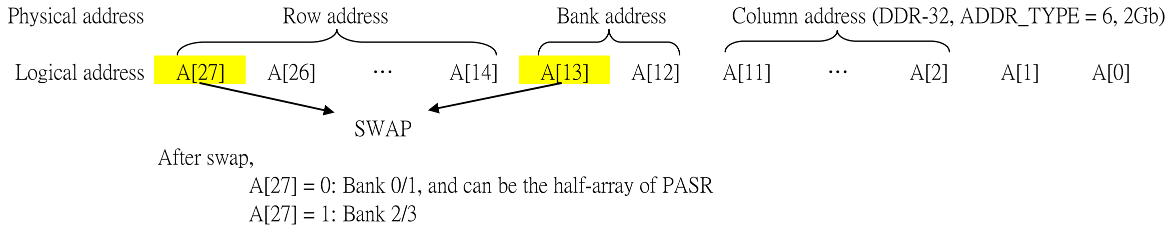
ARF2_EN Single auto-refresh-2 enable (for DRAM initialize)

LDMR_EN Single load mode register enable (for DRAM initialize)

LDEM_EN Single load extended mode register enable (for DRAM initialize)

ADDR_SWAP Swap MSB of DRAM row address and BA[1] for Partial Array Self Refresh (PASR)

- **0** Disable
- **1** Enable



Logical address: [27:0]

ADDR_TYPE	Row address bits	Bank address bits	Column address bits	SDR (DDR) 16-bit	SDR (DDR) 32-bit
010	12	2	8	64Mb (22 <-> 10)	128Mb (23 <-> 11)
011	12	2	9	128Mb (23 <-> 11)	256Mb (24 <-> 12)
100	13	2	9	256Mb (24 <-> 11)	512Mb (25 <-> 12)
101	13	2	10	512Mb (25 <-> 12)	1Gb (26 <-> 13)
110	14	2	10	1Gb (26 <-> 12)	2Gb (27 <-> 13)

ADDR_TYPE DRAM address type

ADDR_TYPE	Row address	Bank	Column	SDR (DDR)	SDR (DDR)
010	12	2	8	64Mb	128Mb
011	12	2	9	128Mb	256Mb
100	13	2	9	256Mb	512Mb
101	13	2	10	512Mb	1Gb
110	14	2	10	1Gb	2Gb

(Others are reserved)

DRAM_TYPE DRAM type selection (SDR/DDR, data bus width 16/32 bits)

- **00** SDR-16
- **01** SDR-32
- **10** DDR-16
- **11** DDR-32

DQSI_CAL_MON Select CS DQS calibrating value for F/W read out

00 Select CS[0] DQS calibrating value for F/W read out
01 Select CS[1] DQS calibrating value for F/W read out
10 Select CS[2] DQS calibrating value for F/W read out
11 Select CS[3] DQS calibrating value for F/W read out

CAL_DONE Digital DLL calibration status
0 Not finish or calibration is disabled
1 Done

CAL_EN Enable digital DLL calibration circuit
0 Disable
1 Enable

SREF_ST DRAM self refresh status
0 Exit self refresh status
1 In self refresh status

PDN_ST DRAM power down status
0 Exit power down status
1 In power down status

SREF_EN DRAM self-refresh enable by AP side. Both SREF_EN and SREF_EN_MD (EMI_MDCL[0]) should be enabled for entering self-refresh mode.
0 Force DRAM to enter self refresh
1 Force DRAM to exit self refresh

PDN_EN Enable DRAM to enter power down mode when DRAM controller is idle (the controller will exit power down mode, and exercise auto refresh step to keep data correctness in DRAM, if the refresh time is reached)
0 Disable
1 Enable

REF_FIX_CK Enable auto refresh with a fixed clock source (3.25MHz) for the refresh counter. Note that REF_CNT_EN (EMI_CONN[1]) is ignored when this bit is enabled!
0 Refresh counter is by HCLK_CK
1 Refresh counter is by a fixed clock 3.25MHz (thus the clock switching will not affect the refresh rate to DRAM!)

REF_CNT_EN Enable auto refresh to DRAM. Note that this bit is ignored when REF_FIX_CK (EMI_CONN[2]) is enabled!
0 Disable
1 Enable

DRAM_EN Enable DRAM controller
0 Disable
1 Enable

+0070h EMI general control registers A EMI_GENA

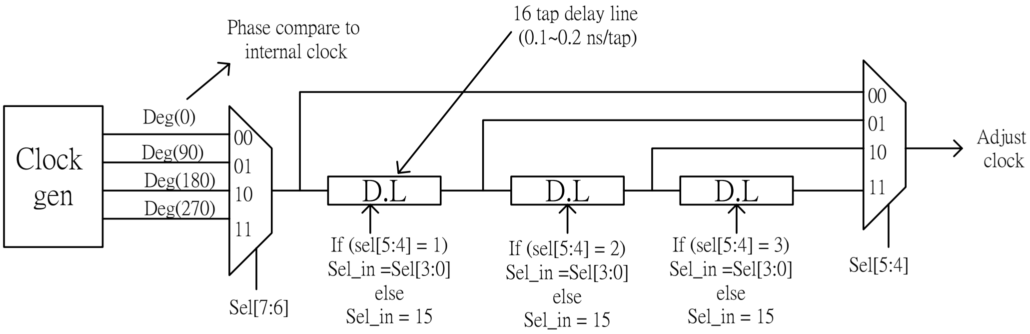
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SW_PSEL**								**SR_PSEL**							
Type	R/W								R/W							
Reset	0								0							

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HI_PIO				TRAS_INTER_DIS	SCLK_EN	DCLK_EN	HCLKX2_CK_ON	CRE_EN	CRE_VALUE	ACTIVE_WR_DIS	ACTIVE_RD_DIS	PAUSE_STR_EN	RWWR_INTER_DIS	RM1	RM0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SW_PSEL Define EMI output clock to PSRAM phase select

SW_PSEL[7:0] : Adjust phase delay --- > 1 tape (0.1~0.2 ns)

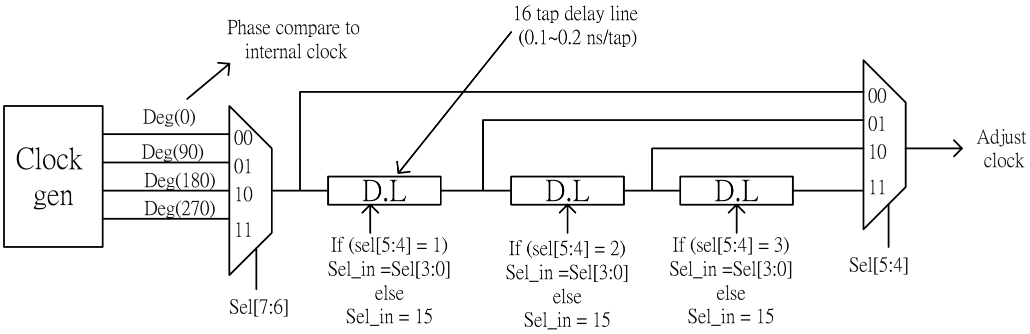
SW_PSEL is only adjustable at SW_DEFAULT = 0



SR_PSEL Define EMI internal adjustable clock to sample (ADMUX & AD-DeMUX) PSRAM input data

SR_PSEL[7:0] : Adjust phase delay --- > 1 tape (0.1~0.2 ns)

SR_PSEL is not adjustable at DATA_A_LAT = 0 and DEMUX_2T_DV = 0 (ADMUX_2T_DV = 0)



HI_PIO Arbitration high priority for AHB 6 ~ 3 (AHB6~5 from CEVA, AHB4~3 from MD)

0 Disable

1 Enable

TRAS_INTER_DIS DRAM inter-bank access is affected by the RAS_MIN_CYC condition

0 Not affected

1 Affected

SCLKEN SRAM controller clock out enable

0 Disable

1 Enable

DCLKEN DRAM controller clock out enable

0 Disable

1 Enable

HCLKX2_CK_ON EMI delay-line input HCLKX2_CK enable. Note that this bit must be enabled under burst-mode PSRAM/NOR and DRAM.

- **0** Disable
- **1** Enable

CRE_EN Assign EA26 as GPIO function for PSRAM CRE

- **0** Disable
- **1** Enable

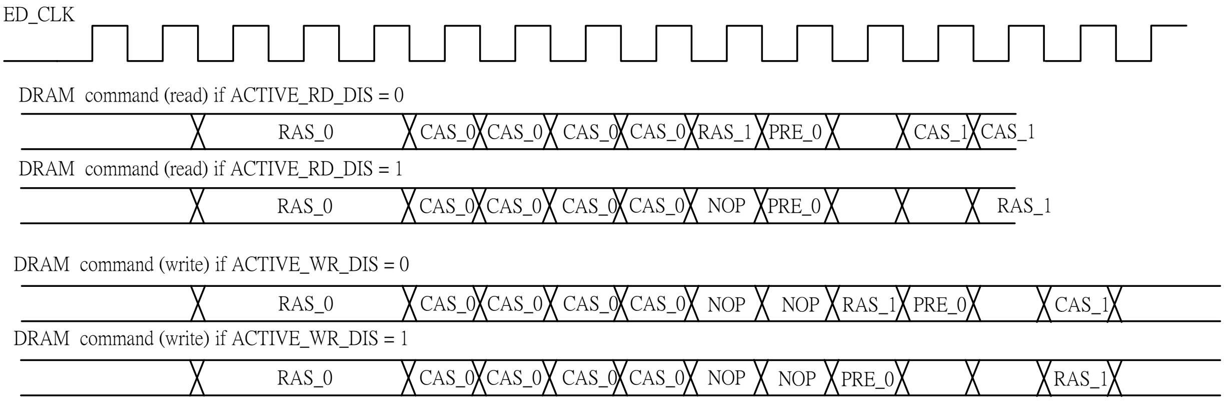
CRE_VALUE Assign CRE output value

ACTIVE_WR_DIS DRAM inter-bank access for write

- **0** Enable
- **1** Disable

ACTIVE_RD_DIS DRAM inter-bank access for read

- **0** Enable
- **1** Disable



PAUSE_STR_EN EMI DRAM controller enters self-refresh controlled by bus pause-start signal. When pause-start signal is asserted, EMI DRAM controller automatically enters self-refresh mode. When pause-start signal is de-asserted, EMI DRAM controller automatically exits self-refresh mode. Pause-start signal is asserted only when AP side and MD side sleep controller are both activated.

- **0** Disable
- **1** Enable

RWWR_INTER_DIS DRAM inter-bank access for read to write and write to read

- **0** Enable
- **1** Disable

RM1 Booting control

- **0** Internal boot
- **1** External boot

When internal boot (RM1 = 0) is selected, ARM will fetch 2 fixed instructions from EMI and jump into the boot ROM area. During the boot ROM execution, RM1 must be set to 1 before burst transactions to EMI!

RM0 Chip select remapping control

0 CS[0]/CS[1] not change
1 CS[0]/CS[1] change

+0078h EMI IO control registers B EMI_GENB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			DQS_SMT	DQ_SMT	DQS_PU	DQS_PD	DQ_PU	DQ_PD	DQS3SR	DQS3E2	DQS3E4	DQS3E8	DQS20SR	DQS2E2	DQS2E4	DQS2E8
Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQS1SR	DQS1E2	DQS1E4	DQS1E8	DQS0SR	DQS0E2	DQS0E4	DQS0E8	DCKSR	DCKE2	DCKE4	DCKE8	SCKSR	SCKE2	SCKE4	SCKE8
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

home

DQS_SMT Enable DQS Input control by Schmitt-Trigger

0 Disable
1 Enable

DQ_SMT Enable DQ Input control by Schmitt-Trigger

0 Disable
1 Enable

DQS_PU Enable DQS Pad pull up control

0 Disable
1 Enable

DQS_PD Enable DQS Pad pull down control

0 Disable
1 Enable

DQ_PU Enable DQ Pad pull up control

0 Disable
1 Enable

DQ_PD Enable DQ Pad pull down control

DQS3SR : DQS3 Pad slew rate control

0 Fast
1 Slow

DQS3Ex DQS3 Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

DQS2SR DQS2 Pad slew rate control

0 Fast
1 Slow

DQS2Ex DQS2 Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

DQS1SR DQS1 Pad slew rate control

- **0** Fast
- **1** Slow

DQS1Ex DQS1 Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

DQS0SR DQS0 Pad slew rate control

- **0** Fast
- **1** Slow

DQS0Ex DQS0 Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

DCKSR DCK Pad slew rate control

- **0** Fast
- **1** Slow

DCKEx DCK Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

SCKSR SCK Pad slew rate control

- **0** Fast
- **1** Slow

SCKEx SCK Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

+0080h EMI IO control registers C EMI_GENC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EASR	EAE2	EAE4	EAE8	EDSR	EDE2	EDE4	EDE8
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11		9	8	7	6		4	3	2	1	0
Name	ECSSR	ECSE2	ECSE4	ECSE8	ERWSR	ERWE2	ERWE4	ERWE8	EADVSR	EADVE2	EADVE4	EADVE8	ERCSR	ERCE2	ERCE4	ERCE8
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EASR Address Pad slew rate control

- **0** Fast
- **1** Slow

EAEx Address Pad driving Control

E2: add 2mA, E4: add 4mA, E8: add 8mA.
If all bits (E2~E8) = 0, the pad has basic 2mA driving.

EDSR Data Pad slew rate control

0 Fast

1 Slow

EDEx Data Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

ECSSR CS Pad slew rate control

0 Fast

1 Slow

ECSEx CS Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

ERWSR RD/WR Pad slew rate control

0 Fast

1 Slow

ERWEx RD/WR Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

EADVSR ADV Pad slew rate control

0 Fast

1 Slow

EADVEx ADV Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

ERCSR RAS/CAS Pad slew rate control

0 Fast

1 Slow

ERCEx RAS/CAS Pad driving control

E2: add 2mA, E4: add 4mA, E8: add 8mA.

If all bits (E2~E8) = 0, the pad has basic 2mA driving.

+0088h EMI bank select registers EMI_GEND

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													DRAM_CS_EN			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SRAM_CS_EN			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

DRAM_CS_EN From bank_3 to bank _0 (all banks can be turned on, except the banks assigned to SRAM)

 © 2009 MediaTek Inc.

0 Disable
1 Enable

SRAM_CS_EN From bank _3 to bank _0 (all banks can be turned on, except the banks assigned to DRAM)

0 Disable
1 Enable

+0090h Modem side master address offset registers EMI_GENE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MD_ADDR_OFFSET															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD_ADDR_OFFSET Modem side master offset address

+0098h EMI delay control registers A EMI_DELA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ31_OUT_DEL				DQ30_OUT_DEL				DQ29_OUT_DEL				DQ28_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ27_OUT_DEL				DQ26_OUT_DEL				DQ25_OUT_DEL				DQ24_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00A0h EMI delay control registers B EMI_DELB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ23_OUT_DEL				DQ22_OUT_DEL				DQ21_OUT_DEL				DQ20_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ19_OUT_DEL				DQ18_OUT_DEL				DQ17_OUT_DEL				DQ16_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00A8h EMI delay control registers C EMI_DELC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ15_OUT_DEL				DQ14_OUT_DEL				DQ13_OUT_DEL				DQ12_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ11_OUT_DEL				DQ10_OUT_DEL				DQ9_OUT_DEL				DQ8_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00B0h EMI delay control registers D EMI_DELD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ7_OUT_DEL				DQ6_OUT_DEL				DQ5_OUT_DEL				DQ4_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ3_OUT_DEL				DQ2_OUT_DEL				DQ1_OUT_DEL				DQ0_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00B8h EMI delay control registers E EMI_DELE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ31_IN_DEL				DQ30_IN_DEL				DQ29_IN_DEL				DQ28_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ27_IN_DEL				DQ26_IN_DEL				DQ25_IN_DEL				DQ24_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00C0h EMI delay control registers F EMI_DELF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ23_IN_DEL				DQ22_IN_DEL				DQ21_IN_DEL				DQ20_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ19_IN_DEL				DQ18_IN_DEL				DQ17_IN_DEL				DQ16_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00C8h EMI delay control registers G EMI_DELG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ15_IN_DEL				DQ14_IN_DEL				DQ13_IN_DEL				DQ12_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ11_IN_DEL				DQ10_IN_DEL				DQ9_IN_DEL				DQ8_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00D0h EMI delay control registers H EMI_DELH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ7_IN_DEL				DQ6_IN_DEL				DQ5_IN_DEL				DQ4_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ3_IN_DEL				DQ2_IN_DEL				DQ1_IN_DEL				DQ0_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00D8h EMI delay control registers I EMI_DELI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQS3_OUT_DEL				DQS2_OUT_DEL				DQS1_OUT_DEL				DQS0_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQS3_IN_DEL				DQS2_IN_DEL				DQS1_IN_DEL				DQS0_IN_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00E0h EMI delay control registers J EMI_DELJ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DQS_IN_ADD4	EDCLK_OUT_DEL								RA_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQM3_OUT_DEL				DQM2_OUT_DEL				DQM1_OUT_DEL				DQM0_OUT_DEL			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DQS_IN_ADD4 When 1/5T DLL locked value (EMI_CONM[3:1]) is 3'b111, DQSx_IN_DEL[2] (EMI_DELI) values would set to 1. Thus the DQSx_IN_DEL values would add 4 if their original DQSx_IN_DEL[2] is 0.

0 Disable

1 Enable

DQX_OUT_DEL (31~0) DDR output data delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

DQX_IN_DEL (31~0) DDR input data delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

DQSX_OUT_DEL (3~0) DDR output data strobe delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

DQSX_IN_DEL (3~0) DDR input data strobe delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance.

Note that when DQS_IN_ADD4 (EMI_DELJ[28]) is enabled and DIGITAL_DLL_CAL_VALUE[3:1] (EMI_CONM[3:1]) is locked to full (3'b111), DQSx_IN_DEL[3:0] values will add 4 if their original DQSx_IN_DEL[2] is 0. The modified values of DQSx_IN_DEL can also be read out by SW.

EDCLK_OUT_DEL DRAM output clock delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

RA_OUT_DEL DDR output address delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

DQMX_OUT_DEL (3~0) DDR output data mask delay balance select (1tap: 0.1~0.2ns), for substrate or PCB skew balance

+00E8h Memory protect unit control registers A EMI_MPUA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_0															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_0															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00F0h Memory protect unit control registers B EMI_MPUB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+00F8h Memory protect unit control registers C EMI_MPUC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_2															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_2															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0100h Memory protect unit control registers D EMI_MPUD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_3															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_3															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0108h Memory protect unit control registers E EMI_MPUE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_4															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_4															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0110h Memory protect unit control registers F EMI_MPUF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_5															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_5															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0118h Memory protect unit control registers G EMI_MPUG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_6															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_6															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0120h Memory protect unit control registers H EMI_MPUH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MPU_START_ADDR_7															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_STOP_ADDR_7															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MPU_START_ADDR_X Memory protect region X start address. {MPU_START_ADDR_X, 14'b0}, total 30-bit

MPU_STOP_ADDR_X Memory protect region X stop address. {MPU_STOP_ADDR_X, 14'b0}, total 30-bit

+0128h Memory protect unit control registers I EMI_MPUI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MD_R7	MD_R6	MD_R5	MD_R4	MD_R3	MD_R2	MD_R1	MD_R0	MD_W7	MD_W6	MD_W5	MD_W4	MD_W3	MD_W2	MD_W1	MD_W0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AP_R7	AP_R6	AP_R5	AP_R4	AP_R3	AP_R2	AP_R1	AP_R0	AP_W7	AP_W6	AP_W5	AP_W4	AP_W3	AP_W2	AP_W1	AP_W0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

 © 2009 MediaTek Inc.

+0130h Memory protect unit control registers J EMI_MPUJ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									M7_EN	M6_EN	M5_EN	M4_EN	M3_EN	M2_EN	M1_EN	M0_EN
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CV_R7	CV_R6	CV_R5	CV_R4	CV_R3	CV_R2	CV_R1	CV_R0	CV_W7	CV_W6	CV_W5	CV_W4	CV_W3	CV_W2	CV_W1	CV_W0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD_Rx Memory protect region_x (7~0) forbid MD-side read.
- **0** Disable
- **1** Enable

MD_Wx Memory protect region_x (7~0) forbid MD-side write.
- **0** Disable
- **1** Enable

AP_Rx Memory protect region_x (7~0) forbid AP-side read.
- **0** Disable
- **1** Enable

AP_Wx Memory protect region_x (7~0) forbid AP-side write.
- **0** Disable
- **1** Enable

Mx_EN Memory protect region_x (7~0) interrupt .
- **0** Disable
- **1** Enable

CV_Rx Memory protect region_x (7~0) forbid CEVA-side read.
- **0** Disable
- **1** Enable

CV_Wx Memory protect region_x (7~0) forbid CEVA -side write.
- **0** Disable
- **1** Enable

+0138h Memory protect unit control registers K (AP & MD can read) EMI_MPUK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MD_R7V	MD_R6V	MD_R5V	MD_R4V	MD_R3V	MD_R2V	MD_R1V	MD_R0V	MD_W7V	MD_W6V	MD_W5V	MD_W4V	MD_W3V	MD_W2V	MD_W1V	MD_W0V
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AP_R7V	AP_R6V	AP_R5V	AP_R4V	AP_R3V	AP_R2V	AP_R1V	AP_R0V	AP_W7V	AP_W6V	AP_W5V	AP_W4V	AP_W3V	AP_W2V	AP_W1V	AP_W0V
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0140h Memory protect unit control registers L (AP & MD can read) EMI_MPUL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name										MPU_ERROR_MASTER[6:0]						
Type										R						
Reset										0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CV_R7V	CV_R6V	CV_R5V	CV_R4V	CV_R3V	CV_R2V	CV_R1V	CV_R0V	CV_W7V	CV_W6V	CV_W5V	CV_W4V	CV_W3V	CV_W2V	CV_W1V	CV_W0V
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MPU_ERROR_MASTER Memory protect error master 6 ~ 0

MD_Rx Memory protect region_x (7~0) MD-side read violate.

MD_Wx Memory protect region_x (7~0) MD-side write violate.

AP_Rx Memory protect region_x (7~0) AP-side read violate.

AP_Wx Memory protect region_x (7~0) AP-side write violate.

CV_Rx Memory protect region_x (7~0) CEVA-side read violate.

CV_Wx Memory protect region_x (7~0) CEVA-side write violate.

+0148h Memory protect unit control registers M (AP & MD can read) EMI_MPUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			MPU_ERROR_ADDRESS													
Type			R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPU_ERROR_ADDRESS															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MPU_ERROR_ADDRESS Memory protecting error address

+0150h Memory protect unit control registers N (AP & MD can read) EMI_MPUN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															CV_IRQ	AP_IRQ
Type															WC	WC
Reset															0	0

AP_IRQ Memory protect error interrupt to AP side (write clear only by AP).

CV_IRQ Memory protect error interrupt to CEVA side (write clear only by AP).

Note:

1. AP accesses protected area:
   - (a) Issue interrupts to inform AP, MD, CEVA
   - (b) AP & CEVA interrupts are cleared by AP
   - (c) MD interrupt is cleared by MD
2. MD accesses protected area:
   - (a) Issue interrupts to inform MD, AP
   - (b) MD interrupt is cleared by MD
   - (c) AP interrupt is cleared by AP
3. CEVA accesses protected area:
   - (a) Issue interrupts to inform CEVA, AP
   - (b) Both interrupts are cleared by AP

+0158h Memory protect unit control registers O (AP & MD can read) EMI_MPUO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**MD_IRQ**
Type																WC
Reset																0

MD_IRQ Memory protect error interrupt to MD side (write clear only by MD).

Note:

1. AP accesses protected area:
   - (a) Issue interrupts to inform AP, MD, CEVA
   - (b) AP & CEVA interrupts are cleared by AP
   - (c) MD interrupt is cleared by MD
2. MD accesses protected area:
   - (a) Issue interrupts to inform MD, AP
   - (b) MD interrupt is cleared by MD
   - (c) AP interrupt is cleared by AP
3. CEVA accesses protected area:
   - (a) Issue interrupts to inform CEVA, AP
   - (b) Both interrupts are cleared by AP

+0160h EMI bus monitor control registers EMI_BMEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**SEL_MASTER**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	SEL_SAME_BANK	SEL_IDLE_TRANS	SEL_INTER_REF	SEL_INTER_BANK	SEL_DIFF_CS	SEL_DRAM_IDLE		BC_OVERRUN			BUS_MON_RW				BUS_MON_PAUSE	BUS_MON_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W		R			R/W				R/W	R/W
Reset	0	0	0	0	0	0		0			0				0	0

SEL_MASTER Monitor the selected masters (master 0 ~ master 11)

EMI_BMEN[16]: Dcache
EMI_BMEN[17]: AP DMA
EMI_BMEN[18]: Icache
EMI_BMEN[19]: MD L1 cache
EMI_BMEN[20]: MD DMA
EMI_BMEN[21]: CEVA1
EMI_BMEN[22]: CEVA2
EMI_BMEN[23]: GMC1 1 (from graph1)
EMI_BMEN[24]: GMC1 2 (from graph1)
EMI_BMEN[25]: GMC2 1 (from graph2)
EMI_BMEN[26]: GMC2 2 (from graph2)
EMI_BMEN[27]: Dummy read

SEL_SAME_BANK Select WORD_ALL_CNT/SAME_BANK_CNT when 0/1

SEL_IDLE_TRANS Select WORD_CNT/IDLE_TRANS_CNT when 0/1

SEL_INTER_REF Select BUSY_ALL_CNT/INTER_REF_CNT when 0/1

SEL_INTER_BANK Select BUSY_CNT/INTER_BANK_CNT when 0/1

SEL_DIFF_CS Select BUSCYC_CNT/DIFF_CS_CNT when 0/1

Note that

TRANS_ALL_CNT = SAME_BANK_CNT + IDLE_TRANS_CNT + INTER_REF_CNT + INTER_BANK_CNT +
DIFF_CS_CNT

SEL_DRAM_IDLE Select TRANS_CNT/DRAM_IDLE_CNT when 0/1

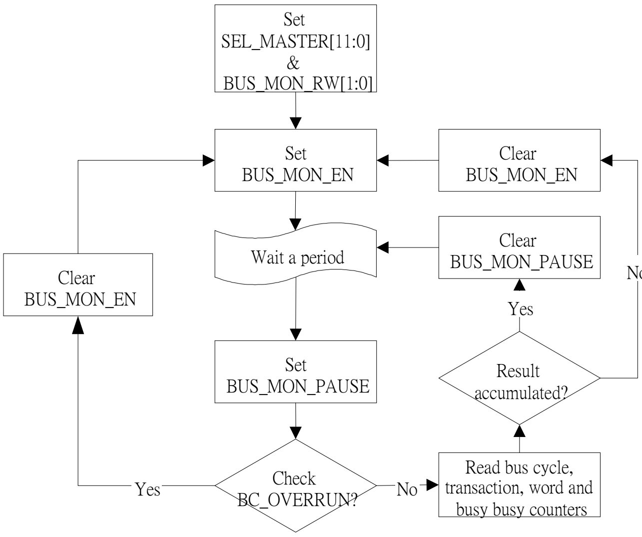
BC_OVERRUN Bus counter (BUSCYC_CNT[31:0]) overrun, and it is cleared by (BUS_MON_EN = 0)

BUS_MON_RW Bus monitor for read/write

- **00** Monitor both read/write transactions
- **01** Monitor only read transactions
- **10** Monitor only write transactions
- **11** Monitor both read/write transactions

BUS_MON_PAUSE Pause the monitor circuit

BUS_MON_EN Enable the monitor circuit. When disable, all monitor counters will be cleared



+0168h EMI bus cycle counters EMI_BCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUSCYC_CNT [31:16] / DIFF_CS_CNT[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSCYC_CNT [15:0] / DIFF_CS_CNT[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BUSCYC_CNT / DIFF_CS_CNT Bus cycle counter, and will keep to maximum when reach / Counters for number of different CS transaction, and will keep to maximum when reach. The counter selection is by EMI_BMEN[11] (SEL_DIFF_CS).

+0170h EMI total transaction counters EMI_TACT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TRANS_ALL_CNT [31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TRANS_ALL_CNT [15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRANS_ALL_CNT Counter for transactions of all masters (master 0 ~ master 11), and will keep to maximum when reach

+0178h EMI transaction counters EMI_TSCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TRANS_ CNT [31:16] / DRAM_IDLE_CNT[31:16]															
Type	R															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TRANS_ CNT [15:0] / DRAM_IDLE_CNT[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRANS_CNT / DRAM_IDLE_CNT Counter for transactions of selected masters (master 0 ~ master 11), and will keep to maximum when reach / Counters for idle cycles of DRAM data bus, and will keep to maximum when reach. The counter selection is by EMI_BMEN[10] (SEL_DRAM_BUSY).

+0180h EMI total word counters EMI_WACT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WORD_ALL_CNT [31:16] / SAME_BANK_CNT[31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WORD_ALL_CNT [15:0] / SAME_BANK_CNT[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WORD_ALL_CNT / SAME_BANK_CNT Counter for access amount (Unit: double words, 8-byte) to EMI of all masters (master 0 ~ master 11), and will keep to maximum when reach / Counters for number of same-bank transaction, and will keep to maximum when reach. The counter selection is by EMI_BMEN[15] (SEL_SAME_BANK).

+0188h EMI word counters EMI_WSCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WORD_ CNT [31:16] / IDLE_TRANS_CNT[31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WORD_ CNT [15:0] / IDLE_TRANS_CNT[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WORD_CNT / IDLE_TRANS_CNT Counter for access amount (Unit: double words, 8-byte) to EMI of selected masters (master 0 ~ master 11), and will keep to maximum when reach / Counters for number of idle to read/write transaction, and will keep to maximum when reach. The counter selection is by EMI_BMEN[14] (SEL_IDLE_TRANS).

+0190h EMI total access cycle counters EMI_BACT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUSY_ALL_CNT [31:16] / INTER_REF_CNT[31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY_ALL_CNT [15:0] / INTER_REF_CNT[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2009 MediaTek Inc.

BUSY_ALL_CNT / INTER_REF_CNT Counter for access cycles to EMI of all masters (master 0 ~ master 11), and will keep to maximum when reach / Counters for number of inter-bank access blocked by refresh cycles, and will keep to maximum when reach. The counter selection is by EMI_BMEN[13] (SEL_INTER_REF).

+0198h EMI access cycle counters EMI_BSCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUSY_CNT [31:16] / INTER_BANK_CNT[31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY_CNT [15:0] / INTER_BANK_CNT[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BUSY_CNT / INTER_BANK_CNT Counter for access cycles to EMI of selected masters (master 0 ~ master 11) / Counters for number of inter-bank transactions to DRAM bus, and will keep to maximum when reach. The counter selection is by EMI_BMEN[12] (SEL_INTER_BANK)

+01A0h EMI dummy read control registers EMI_DRCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			DUMMY_RD_ADDR													
Type			R/W													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DUMMY_RD_PERIOD			DUMMY_RD_BURST					DUMMY_RD_SIZE		DR_TEST_MODE		DR_CS_TOGGLE	DUMMY_RD_EN
Type			R/W		R/W	R/W					R/W		R/W		R/W	R/W
Reset			0		0	0					0		0		0	0

DUMMY_RD_ADDR Dummy read address {DUMMY_RD_ADDR[29:16], 16'b0}. The CS address ADDR[29:28] will be sequentially toggled if DR_CS_TOGGLE is enabled.

DUMMY_RD_PERIOD When DR_TEST_MODE is disabled, the dummy read period, 0/1/2/3 = 1.288/2.576/5.152/10.304 seconds when EMI clock is 104MHz.
Note that when DR_CS_TOGGLE is enabled, the dummy read will issue to each installed rank (EMI_GEND[19:16], DRAM_SC_EN) sequentially according to this period setting. Thus the dummy read period to each rank will be extended if multiple ranks are installed!

DUMMY_RD_BURST Burst type of AHB protocol for dummy read ('b001 INCR is not supported!)

000 SINGLE
001 Reserved
010 WRAP 4 (suggest)
011 INCR 4
100 WRAP 8
101 INCR 8
110 WRAP 16
111 INCR 16

DUMMY_RD_SIZE Transfer size of AHB protocol for dummy read

00 1-byte

01 2-byte

10 4-byte (suggest)

11 8-byte

DR_TEST_MODE When enabled, the dummy read period (DUMMY_RD_PERIOD) is reduced to 0/1/2/3 = 10.24/20.48/40.96/81.92 us when EMI clock is 104MHz

DR_CS_TOGGLE Toggle DRAM CS for dummy read transaction

0 Dummy read CS address is by DUMMY_RD_ADDR

1 Dummy read CS address will sequentially issue to installed DRAM CS's (EMI_GEND[19:16], DRAM_SC_EN)

DUMMY_RD_EN Enable the dummy read function. When EMI enters self-refresh mode, Dummy Read function is automatically disabled. Manually turn off EMI_DRCT[0]: DUMMY_RD_EN before enter self-refresh mode is unnecessary.

+01B0h ~ 01C8 EMI DQS auto tracking control registers EMI_DQSA ~ D

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQSI3_CAL_ENABLE	DQSI3_DLYSEL							DQSI2_CAL_ENABLE	DQSI2_DLYSEL						
Type	R/W	R/W							R/W	R/W						
Reset	0	0							0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQSI1_CAL_ENABLE	DQSI1_DLYSEL							DQSI0_CAL_ENABLE	DQSI0_DLYSEL						
Type	R/W	R/W							R/W	R/W						
Reset	0	0							0	0						

EMI_DQSA for CS[0], EMI_DQSB for CS[1], EMI_DQSC for CS[2] and EMI_DQSD for CS[3]

DQSIx_CAL_ENABLE Enable auto-tracking function for input DQS[x]

DQSIx_DLYSEL[6:0] Delay selection of auto-tracking function for input DQS[x]

DQSIx_DLYSEL[6:5]: 00/01/10/11 = Delay 0T/1T/2T/3T of hclk_ck (1X clock)

DQSIx_DLYSEL[4]: 0/1 = Delay 0T/1T of hclkx2_ck (2X clock)

DQSIx_DLYSEL[3]: 0/1 = Delay 0T/0.5T of hclkx2_ck (2X clock)

DQSIx_DLYSEL[2:0]: Delay 0.3ns/1Tap at typical case

+01D0h EMI DQS auto tracking value registers EMI_DQSV

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	0	DQSI3_DLYSEL_CAL							0	DQSI2_DLYSEL_CAL						
Type	R	R							R	R						
Reset	0	0							0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	DQSI1_DLYSEL_CAL							0	DQSI0_DLYSEL_CAL						

Type	R	R	R	R
Reset	0	0	0	0

DQSIx_DLYSEL_CAL[6:0] Calibrating delay value of auto-tracking function for input DQS[x], the corresponding CS is selected by EMI_CONN[15:14] (DQSI_CAL_MON)
DQSIx_DLYSEL_CAL[6:5]: 00/01/10/11 = Delay 0T/1T/2T/3T of hclk_ck (1X clock)
DQSIx_DLYSEL_CAL [4]: 0/1 = Delay 0T/1T of hclkx2_ck (2X clock)
DQSIx_DLYSEL_CAL [3]: 0/1 = Delay 0T/0.5T of hclkx2_ck (2X clock)
DQSIx_DLYSEL_CAL [2:0]: Delay 0.3ns/1Tap at typical case

+01E0h EMI modem side control registers EMI_MDCL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SREF_ST_MD	SREF_EN_MD
Type															R	R/W
Reset															0	0

SREF_ST_MD DRAM self refresh status by MD read
- **0** Not in self refresh
- **1** In self refresh

SREF_EN_MD DRAM self-refresh enable by modem side. Both SREF_EN (EMI_CONN[5]) and SREF_EN_MD should be enabled for entering self-refresh mode.
- **0** Force DRAM to exit self refresh
- **1** Force DRAM to enter self refresh

2.14.3 1.1 EMI AHB & GMC Bus Monitor

In order to evaluate MT6516 system performance, this EMI AHB & GMC bus monitor is added to observe EMI front-end AHB & GMC bus behavior and record some useful data, such as latency, data amount, and so on. This monitor locates at stage2 in the following figure. EMI AHB bus totally has 7 ports, including 3 ports from APMCUSYS, 2 ports from MDMCUSYS, and 2 ports from CEVASYS, while GMC bus has four ports. This monitor only can observe one port at a time.

Figure1. MT6516 Performance Monitor Overview

REGISTER ADDRESS	REGISTER NAME	SYNONYM
8003_7000h	EMI Bus Monitor Enable Register	EMIMON_EN
8003_7004h	EMI Bus Monitor Latch Register	EMIMON_LATCH
8003_7008h	EMI Bus Monitor Latch Clear Register	EMIMON_LATCH_CLR
8003_700Ch	EMI Bus Monitor Read/Write Selection Register	EMIMON_RW

8003_7010h	EMI Bus Monitor CP/DP Selection Register	EMIMON_MAX_SEL
8003_7014h	EMI Bus Monitor Idle Mode Selection Register	EMIMON_ISEL
8003_7018h	EMI Bus Monitor Source Selection Register	EMIMON_SRCSEL
8003_701Ch	EMI Bus Monitor Cycle Count Register	EMIMON_CYCCNT
8003_7020h	EMI Bus Monitor Max Latency Register	EMIMON_MAXLAT
8003_7024h	EMI Bus Monitor Max Command/Data Phase Register	EMIMON_MAXCPDP
8003_7028h	EMI Bus Monitor Command Phase Accumulation Register	EMIMON_CP
8003_702Ch	EMI Bus Monitor Data Phase Accumulation Register	EMIMON_DP
8003_7030h	EMI Bus Monitor Idle Cycle Count Register	EMIMON_IDLE
8003_7034h	EMI Bus Monitor Request Number Register	EMIMON_REQ
8003_7038h	EMI Bus Monitor Data Beat Register	EMIMON_DBT
8003_703Ch	EMI Bus Monitor Data Byte Register	EMIMON_DBYTE
8003_7040h	EMI Bus Monitor Ultra Request Count Register	EMIMON_ULTRA
8003_7044h	EMI Bus Monitor Clear Register	EMIMON_CLR

Table 34 ***APB Register Map***

2.14.4 1.1.1 Register Definition

8003_7000h EMI Bus Monitor Enable Register EMIMON_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																BUS_MON_EN
Type																R/W
Reset																0

BUS_MON_EN : Enable EMI bus monitor

8003_7004h EMI Bus Monitor Latch Register EMIMON_LATCH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																LATCH
Type																WC

Reset																0

LATCH : Write this register no matter what value will perform latch operation to store counter value into register

8003_7008h EMI Bus Monitor Latch Clear Register EMIMON_LATCH_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																LATCH_CLR
Type																R/W
Reset																0

LATCH_CLR : Indicate while latch command is performed also clear counter value

8003_700Ch EMI Bus Monitor Read/Write Selection Register EMIMON_RW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RW_IND		
Type														R/W		
Reset														0		

RW_IND : Indicate read or write or both command is monitored

3’b011 : Read

3’b100 : Write

3’b001 : Read & Write

8003_7010h EMI Bus Monitor CP/DP Selection Register EMIMON_MAX_SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CPDP
Type																R/W
Reset																0

CPDP : Indicate command phase or data phase maximal duration should recorded

0 : data phase

1 : command phase

© 2009 MediaTek Inc.

8003_7014h EMI Bus Monitor Idle Mode Selection Register EMIMON_ISEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ISEL**
Type																R/W
Reset																0

ISEL : Indicate data phase should be counted in idle mode or not

0 : Count in

1 : Not count in

8003_7018h EMI Bus Monitor Source Selection Register EMIMON_SRCSEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**SRCSEL**												
Type				R/W												
Reset				0												

SRCSEL : Indicate which EMI port is selected

12’h1 : AP ARM9 D Bus

12’h2 : AP DMA Bus

12’h4 : AP ARM9 I Bus

12’h8 : MD ARM7 I/D Bus

12’h10 : MD DMA Bus

12’h20 : CEVA AHB1 Bus

12’h40 : CEVA AHB2 Bus

12’h80 : First GMC1 Bus

12’h100 : Second GMC1 Bus

12’h200 : First GMC2 Bus

12’h400 : Second GMC2 Bus

12’h800 : Combined first and second GMC1 Bus

12’h1000 : Combined first and second GMC2 Bus

8003_701Ch EMI Bus Monitor Cycle Count Register EMIMON_CYCCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CYCCNT[31:16]**															
Type	RO															
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CYCCNT[15:0]**															
Type	RO															
Reset	1															

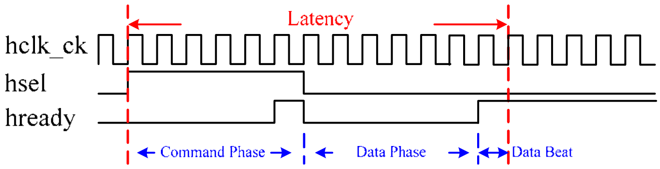
CYCCNT : Bus cycle count in monitor window from start to end. If reach maximal value (32'hffffffff), this register keeps maximal value.

8003_7020h EMI Bus Monitor Max Latency Register EMIMON_MAXLAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MAXLAT[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAXLAT[15:0]**															
Type	RO															
Reset	0															

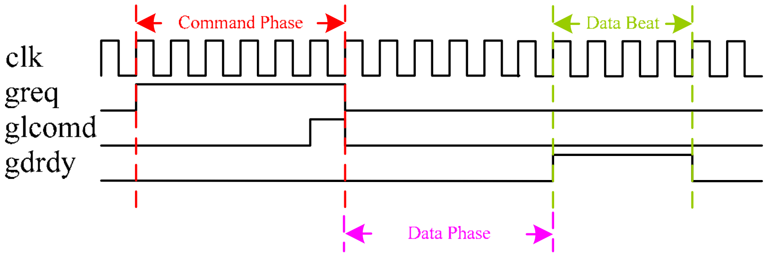
AHB :

Single Read / Write Transaction

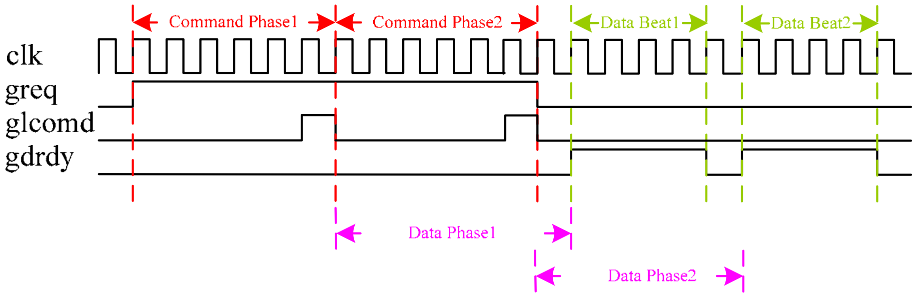


GMC :

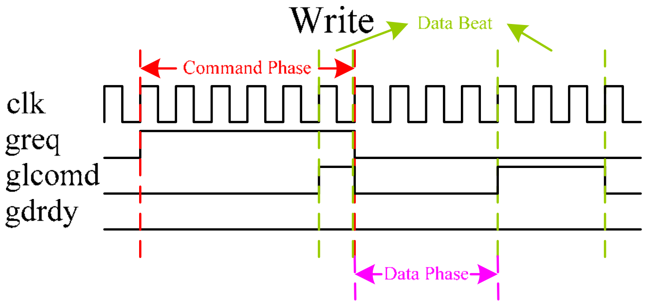
Single Read



Multiple Read



 © 2009 MediaTek Inc.



MAX_LATENCY : This register records maximal latency in one single bus transaction within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value. Latency definition is shown below.

Latency = Command phase + Data phase + Data beat
Command Phase : The cycle count from master starts to issue request to slave accepts command
Data Phase : The cycle count from slave accepts command to slave accepts or returns data
Data Beat : Always 1 in AHB. Each AHB transaction handles one data. In GMC, data beat equals to burst type

8003_7024h EMI Bus Monitor Max Command/Data Phase Register EMIMON_MAXCPDP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MAXCPDP[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MAXCPDP[15:0]															
Type	RO															
Reset	0															

MAXCPDP : This register records maximal command phase or data phase duration within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value.

8003_7028h EMI Bus Monitor Command Phase Accumulation Register EMIMON_CP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CP[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CP[15:0]															
Type	RO															
Reset	0															

CP : This register records total command phase cycles of all transactions within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value.

 © 2009 MediaTek Inc.

8003_702Ch EMI Bus Monitor Data Phase Accumulation Register EMIMON_DP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DP[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DP[15:0]**															
Type	RO															
Reset	0															

DP : This register records total data phase cycles of all transactions within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value.

8003_7030h EMI Bus Monitor Idle Cycle Count Register EMIMON_IDLE

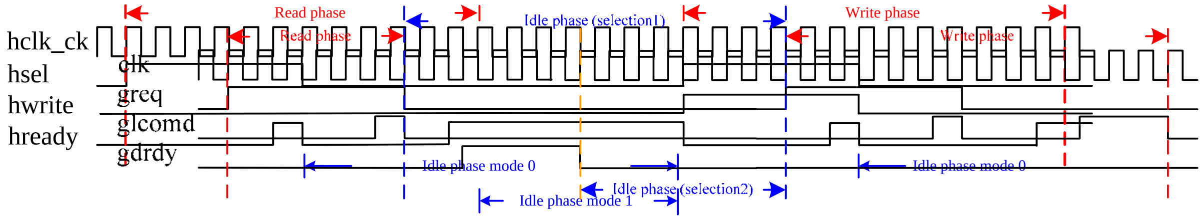
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IDLE_CNT[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IDLE_CNT[15:0]**															
Type	RO															
Reset	0															

IDLE_CNT : This register records total idle cycles within monitor window. If reach maximal value (32'hffffffff), this

register will hold maximal value. Idle cycle means there are no transactions on the bus as shown in the following

figure. Whether data phase counts in idle cycle or not depends on EMIMON_ISEL setting.

AHB :



GMC :

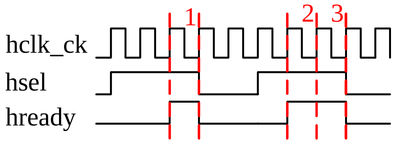
8003_7034h EMI Bus Monitor Request Number Register EMIMON_REQ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**REQ_CNT[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**REQ_CNT[15:0]**															
Type	RO															
Reset	0															

REQ_CNT : This register records total request counts within monitor window. If reach maximal value (32'hffffffff),

this register will hold maximal value. Burst request only counts once though there are more than one transactions in AHB.

This figure is a simple example that 3 requests transmit on the AHB bus.



8003_7038h EMI Bus Monitor Data Beat Register EMIMON_DBT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DBT_CNT[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DBT_CNT[15:0]															
Type	RO															
Reset	0															

DBT_CNT : This register records total data beat counts within monitor window. If reaches maximal value (32'hffffffff), this register will hold maximal value. This value equals to transaction burst type summation.

8003_703Ch EMI Bus Monitor Data Byte Register EMIMON_DBYTE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DBYTE_CNT[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DBYTE_CNT[15:0]															
Type	RO															
Reset	0															

DBYTE_CNT : This register records total data byte counts within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value.

8003_7040h EMI Bus Monitor Ultra Request Count Register EMIMON_ULTRA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ULTRA_CNT[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ULTRA_CNT[15:0]															
Type	RO															
Reset	0															

ULTRA_CNT : This register records GMC ultra request count within monitor window. If reach maximal value (32'hffffffff), this register will hold maximal value.

© 2009 MediaTek Inc.

8003_7044h EMI Bus Monitor Clear Register EMIMON_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**CLR**
Type																R/W
Reset																0

CLR : While assert, clear all counter value

8003_7800h EMI Bus Monitor ARM Liniter EMIMON_LIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**LIMIT_REQ**	
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LIMIT_REQ**									**LIMIT_CYC**						**LIMIT_EN**
Type	R/W									R/W						R/W
Reset	0									0						0

Limit the corresponding target
mom_srcsel==0 => ARM9D
mom_srcsel==5 => CEVA
mom_srcsel==6 => CEVA
it will limit target's request number must be less than {limit_req,7'b0} every {limit_cyc,12'b0} cycles

2.15 General Purpose Inputs/Outputs

MT6516 offers 147 general-purpose I/O pins. By setting the control registers, MCU software can control the direction, the output value, and read the input values on these pins. These GPIOs and are multiplexed with other functionalities to reduce the pin count.

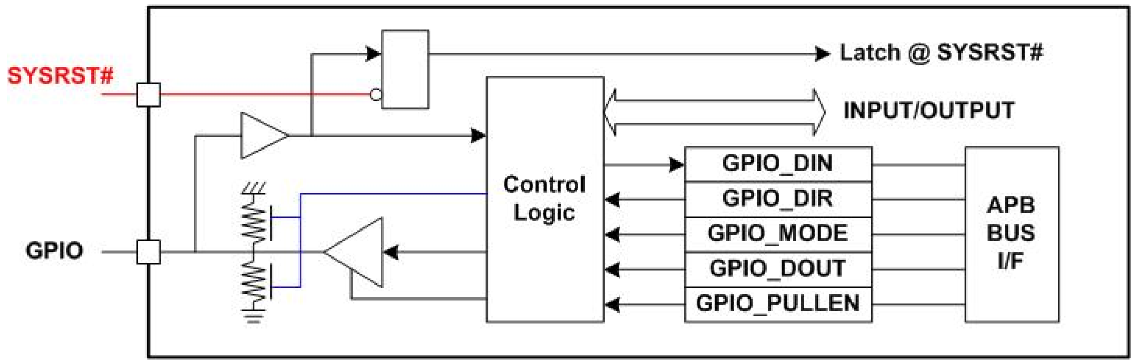


Figure 18 ***GPIO Block Diagram***

GPIOs at RESET

Upon a hardware reset (SYSRST#), GPIOs are all configured as inputs and the following alternative usages of the GPIO pins are enabled.
These GPIOs are used to latch the inputs upon reset to memorize the desired configuration to ensure that the system restarts or boots up in the right mode.

2.15.1 Register Definitions

8000_2000h GPIO direction control register 1 GPIO_DIR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2010h GPIO direction control register 2 GPIO_DIR2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2020h GPIO direction control register 3 GPIO_DIR3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2030h GPIO direction control register 4 GPIO_DIR4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2040h GPIO direction control register 5 GPIO_DIR5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2050h GPIO direction control register 6 GPIO_DIR6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88	GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2060h GPIO direction control register 7 GPIO_DIR7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO111**	**GPIO110**	**GPIO109**	**GPIO108**	**GPIO107**	**GPIO106**	**GPIO105**	**GPIO104**	**GPIO103**	**GPIO102**	**GPIO101**	**GPIO100**	**GPIO99**	**GPIO98**	**GPIO97**	**GPIO96**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2070h GPIO direction control register 8 GPIO_DIR8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO127**	**GPIO126**	**GPIO125**	**GPIO124**	**GPIO123**	**GPIO122**	**GPIO121**	**GPIO120**	**GPIO119**	**GPIO118**	**GPIO117**	**GPIO116**	**GPIO115**	**GPIO114**	**GPIO113**	**GPIO112**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2080h GPIO direction control register 9 GPIO_DIR9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO143**	**GPIO142**	**GPIO141**	**GPIO140**	**GPIO139**	**GPIO138**	**GPIO137**	**GPIO136**	**GPIO135**	**GPIO134**	**GPIO133**	**GPIO132**	**GPIO131**	**GPIO130**	**GPIO129**	**GPIO128**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2090h GPIO direction control register 10 GPIO_DIR10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**GPIO146**	**GPIO145**	**GPIO144**
Type														R/W	R/W	R/W
Reset														0	0	0

GPIO*n* GPIO direction control

- **0** GPIOs are configured as input
- 1 GPIOs are configured as output

8000_2100h GPIO pull-up/pull-down enable register 1 GPIO_PULLEN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO15**	**GPIO14**	**GPIO13**	**GPIO12**	**GPIO11**	**GPIO10**	**GPIO9**	**GPIO8**	**GPIO7**	**GPIO6**	**GPIO5**	**GPIO4**	**GPIO3**	**GPIO2**	**GPIO1**	**GPIO0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

8000_2110h GPIO pull-up/pull-down enable register 2 GPIO_PULLEN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2120h GPIO pull-up/pull-down enable register 3 GPIO_PULLEN3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2130h GPIO pull-up/pull-down enable register 4 GPIO_PULLEN4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2140h GPIO pull-up/pull-down enable register 5 GPIO_PULLEN5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2150h GPIO pull-up/pull-down enable register 6 GPIO_PULLEN6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88	GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2160h GPIO pull-up/pull-down enable register 7 GPIO_PULLEN7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO111	GPIO110	GPIO109	GPIO108	GPIO107	GPIO106	GPIO105	GPIO104	GPIO103	GPIO102	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8000_2170h GPIO pull-up/pull-down enable register 8 GPIO_PULLEN8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO127	GPIO126	GPIO125	GPIO124	GPIO123	GPIO122	GPIO121	GPIO120	GPIO119	GPIO118	GPIO117	GPIO116	GPIO115	GPIO114	GPIO113	GPIO112
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

 © 2009 MediaTek Inc.

8000_2180h GPIO pull-up/pull-down enable register 9 GPIO_ PULLEN 9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO143**	**GPIO142**	**GPIO141**	**GPIO140**	**GPIO139**	**GPIO138**	**GPIO137**	**GPIO136**	**GPIO135**	**GPIO134**	**GPIO133**	**GPIO132**	**GPIO131**	**GPIO130**	**GPIO129**	**GPIO128**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

8000_2190h GPIO pull-up/pull-down enable register 10 GPIO_ PULLEN 10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**GPIO146**	**GPIO145**	**GPIO144**
Type														R/W	R/W	R/W
Reset														1	1	1

GPIO*n* GPIO pull-up/pull-down control

8000_2200h GPIO pull-up/pull-down selection register 1 GPIO_PULLSEL 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO15**	**GPIO14**	**GPIO13**	**GPIO12**	**GPIO11**	**GPIO10**	**GPIO9**	**GPIO8**	**GPIO7**	**GPIO6**	**GPIO5**	**GPIO4**	**GPIO3**	**GPIO2**	**GPIO1**	**GPIO0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2210h GPIO pull-up/pull-down selection register 2 GPIO_PULLSEL 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO31**	**GPIO30**	**GPIO29**	**GPIO28**	**GPIO27**	**GPIO26**	**GPIO25**	**GPIO24**	**GPIO23**	**GPIO22**	**GPIO21**	**PGIO20**	**GPIO19**	**GPIO18**	**GPIO17**	**GPIO16**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2220h GPIO pull-up/pull-down selection register 3 GPIO_PULLSEL 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO47**	**GPIO46**	**GPIO45**	**GPIO44**	**GPIO43**	**GPIO42**	**GPIO41**	**GPIO40**	**GPIO39**	**GPIO38**	**GPIO37**	**GPIO36**	**GPIO35**	**GPIO34**	**GPIO33**	**GPIO32**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2230h GPIO pull-up/pull-down selection register 4 GPIO_PULLSEL4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO63**	**GPIO62**	**GPIO61**	**GPIO60**	**GPIO59**	**GPIO58**	**GPIO57**	**GPIO56**	**GPIO55**	**GPIO54**	**GPIO53**	**GPIO52**	**GPIO51**	**GPIO50**	**GPIO49**	**GPIO48**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2240h GPIO pull-up/pull-down selection register 5 GPIO_PULLSEL5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO79**	**GPIO78**	**GPIO77**	**GPIO76**	**GPIO75**	**GPIO74**	**GPIO73**	**GPIO72**	**GPIO71**	**GPIO70**	**GPIO69**	**GPIO68**	**GPIO67**	**GPIO66**	**GPIO65**	**GPIO64**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2250h GPIO pull-up/pull-down selection register 6 GPIO_PULLSEL6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO95**	**GPIO94**	**GPIO93**	**GPIO92**	**GPIO91**	**GPIO90**	**GPIO89**	**GPIO88**	**GPIO87**	**GPIO86**	**GPIO85**	**GPIO84**	**GPIO83**	**GPIO82**	**GPIO81**	**GPIO80**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2260h GPIO pull-up/pull-down selection register 7 GPIO_PULLSEL7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO111**	**GPIO110**	**GPIO109**	**GPIO108**	**GPIO107**	**GPIO106**	**GPIO105**	**GPIO104**	**GPIO103**	**GPIO102**	**GPIO101**	**GPIO100**	**GPIO99**	**GPIO98**	**GPIO97**	**GPIO96**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2270h GPIO pull-up/pull-down selection register 8 GPIO_PULLSEL8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO127**	**GPIO126**	**GPIO125**	**GPIO124**	**GPIO123**	**GPIO122**	**GPIO121**	**GPIO120**	**GPIO119**	**GPIO118**	**GPIO117**	**GPIO116**	**GPIO115**	**GPIO114**	**GPIO113**	**GPIO112**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2280h GPIO pull-up/pull-down selection register 9 GPIO_ PULLSEL9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO143	GPIO142	GPIO141	GPIO140	GPIO139	GPIO138	GPIO137	GPIO136	GPIO135	GPIO134	GPIO133	GPIO132	GPIO131	GPIO130	GPIO129	GPIO128
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

8000_2290h GPIO pull-up/pull-down selection register 10 GPIO_ PULLSEL10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														GPIO146	GPIO145	GPIO144
Type														R/W	R/W	R/W
Reset														0	0	0
Note														PD	PD	PD

GPIO*n* GPIO pull-up/pull-down selection control

- **0** GPIOs pull-down
- **1** GPIOs pull-up

8000_2300h GPIO data inversion control register 1 GPIO_DINV1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2310h GPIO data inversion control register 2 GPIO_DINV2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2320h GPIO data inversion control register 3 GPIO_DINV3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2330h GPIO data inversion control register 4 GPIO_DINV4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2009 MediaTek Inc.

8000_2340h GPIO data inversion control register 5 GPIO_DINV5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2350h GPIO data inversion control register 6 GPIO_DINV6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88	GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2360h GPIO data inversion control register 7 GPIO_DINV7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO111	GPIO110	GPIO109	GPIO108	GPIO107	GPIO106	GPIO105	GPIO104	GPIO103	GPIO102	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2370h GPIO data inversion control register 8 GPIO_DINV8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO127	GPIO126	GPIO125	GPIO124	GPIO123	GPIO122	GPIO121	GPIO120	GPIO119	GPIO118	GPIO117	GPIO116	GPIO115	GPIO114	GPIO113	GPIO112
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2380h GPIO data inversion control register 9 GPIO_ DINV9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO143	GPIO142	GPIO141	GPIO140	GPIO139	GPIO138	GPIO137	GPIO136	GPIO135	GPIO134	GPIO133	GPIO132	GPIO131	GPIO130	GPIO129	GPIO128
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2390h GPIO data inversion control register 10 GPIO_ DINV10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														GPIO146	GPIO145	GPIO144
Type														R/W	R/W	R/W
Reset														0	0	0

GPIOn GPIO inversion control

- **0** GPIOs data inversion disable
- **1** GPIOs data inversion enable

8000_2400h GPIO data output register 1 GPIO_DOUT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2410h GPIO data output register 2 GPIO_DOUT2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2420h GPIO data output register 3 GPIO_DOUT3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2430h GPIO data output register 4 GPIO_DOUT4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2440h GPIO data output register 5 GPIO_ DOUT5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2450h GPIO data output register 6 GPIO_DOUT6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88	GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2460h GPIO data output register 7 GPIO_DOUT7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO111	GPIO110	GPIO109	GPIO108	GPIO107	GPIO106	GPIO105	GPIO104	GPIO103	GPIO102	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2470h GPIO data output register 8 GPIO_DOUT8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO127**	**GPIO126**	**GPIO125**	**GPIO124**	**GPIO123**	**GPIO122**	**GPIO121**	**GPIO120**	**GPIO119**	**GPIO118**	**GPIO117**	**GPIO116**	**GPIO115**	**GPIO114**	**GPIO113**	**GPIO112**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2480h GPIO data output register 9 GPIO_ DOUT9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO143**	**GPIO142**	**GPIO141**	**GPIO140**	**GPIO139**	**GPIO138**	**GPIO137**	**GPIO136**	**GPIO135**	**GPIO134**	**GPIO133**	**GPIO132**	**GPIO131**	**GPIO130**	**GPIO129**	**GPIO128**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8000_2490h GPIO data output register 10 GPIO_ DOUT10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**GPIO146**	**GPIO145**	**GPIO144**
Type														R/W	R/W	R/W
Reset														0	0	0

GPIOn GPIO data output control

- **0** GPIOs data output 0
- **1** GPIOs data output 1

8000_2500h GPIO data Input register 1 GPIO_DIN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO15**	**GPIO14**	**GPIO13**	**GPIO12**	**GPIO11**	**GPIO10**	**GPIO9**	**GPIO8**	**GPIO7**	**GPIO6**	**GPIO5**	**GPIO4**	**GPIO3**	**GPIO2**	**GPIO1**	**GPIO0**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2510h GPIO data Input register 2 GPIO_DIN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO31**	**GPIO30**	**GPIO29**	**GPIO28**	**GPIO27**	**GPIO26**	**GPIO25**	**GPIO24**	**GPIO23**	**GPIO22**	**GPIO21**	**PGIO20**	**GPIO19**	**GPIO18**	**GPIO17**	**GPIO16**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2520h GPIO data Input register 3 GPIO_DIN3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO47**	**GPIO46**	**GPIO45**	**GPIO44**	**GPIO43**	**GPIO42**	**GPIO41**	**GPIO40**	**GPIO39**	**GPIO38**	**GPIO37**	**GPIO36**	**GPIO35**	**GPIO34**	**GPIO33**	**GPIO32**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2530h GPIO data input register 4 GPIO_DIN4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO63**	**GPIO62**	**GPIO61**	**GPIO60**	**GPIO59**	**GPIO58**	**GPIO57**	**GPIO56**	**GPIO55**	**GPIO54**	**GPIO53**	**GPIO52**	**GPIO51**	**GPIO50**	**GPIO49**	**GPIO48**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2540h GPIO data input register 5 GPIO_ DIN5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO79**	**GPIO78**	**GPIO77**	**GPIO76**	**GPIO75**	**GPIO74**	**GPIO73**	**GPIO72**	**GPIO71**	**GPIO70**	**GPIO69**	**GPIO68**	**GPIO67**	**GPIO66**	**GPIO65**	**GPIO64**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2550h GPIO data input register 6 GPIO_DIN6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO95**	**GPIO94**	**GPIO93**	**GPIO92**	**GPIO91**	**GPIO90**	**GPIO89**	**GPIO88**	**GPIO87**	**GPIO86**	**GPIO85**	**GPIO84**	**GPIO83**	**GPIO82**	**GPIO81**	**GPIO80**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2560h GPIO data input register 7 GPIO_DIN7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO111**	**GPIO110**	**GPIO109**	**GPIO108**	**GPIO107**	**GPIO106**	**GPIO105**	**GPIO104**	**GPIO103**	**GPIO102**	**GPIO101**	**GPIO100**	**GPIO99**	**GPIO98**	**GPIO97**	**GPIO96**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2570h GPIO data input register 8 GPIO_DIN8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO127**	**GPIO126**	**GPIO125**	**GPIO124**	**GPIO123**	**GPIO122**	**GPIO121**	**GPIO120**	**GPIO119**	**GPIO118**	**GPIO117**	**GPIO116**	**GPIO115**	**GPIO114**	**GPIO113**	**GPIO112**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2580h GPIO data input register 9 GPIO_ DIN9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO143**	**GPIO142**	**GPIO141**	**GPIO140**	**GPIO139**	**GPIO138**	**GPIO137**	**GPIO136**	**GPIO135**	**GPIO134**	**GPIO133**	**GPIO132**	**GPIO131**	**GPIO130**	**GPIO129**	**GPIO128**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

8000_2590h GPIO data input register 10 GPIO_ DIN10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name														GPIO146	GPIO145	GPIO144
Type														R/W	R/W	R/W
Reset														X	X	X

GPIOn GPIOs data input

8000_2600h GPIO mode control register 1 GPIO_MODE1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO7_M		GPIO6_M		GPIO5_M		GPIO4_M		GPIO3_M		GPIO2_M		GPIO1_M		GPIO0_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO0_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: PWM4
- **10** O: CLK_OUT0
- **11** Reserved

GPIO1_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: EADMUX
- **10** O: CLK_OUT1
- **11** I: External interrupt input 15 (EINT15)

GPIO2_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Reserved
- **10** I2S_WS
- **11** Reserved

GPIO3_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Reserved
- **10** I2S_CLK
- **11** Reserved

GPIO4_M GPO mode selection
- **00** Configured as GPIO function
- **01** Reserved
- **10** I2S_DAT
- **11** Reserved

GPIO5_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Reserved
- **10** Reserved
- **11** Reserved

GPIO6_M GPIO mode selection
- **00** Configured as GPIO function

01 Reserved
10 Reserved
11 Reserved
GPIO7_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

8000_2610h GPIO mode control register 2 GPIO_MODE2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15_M		GPIO14_M		GPIO13_M		GPIO12_M		GPIO11_M		GPIO10_M		GPIO9_M		GPIO8_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO8_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 O: CLK_OUT2
11 Reserved
GPIO9_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 O: CLK_OUT3
11 Reserved
GPIO10_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved
GPIO11_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved
GPIO12_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved
GPIO13_M GPIO mode selection
00 Configured as GPIO function
01 I: CEVA_TCK

10 Reserved
11 Reserved
GPIO14_M GPIO mode selection
00 Configured as GPIO function
01 I: CEVA_TMS
10 Reserved
11 Reserved
GPIO15_M GPIO mode selection
00 Configured as GPIO function
01 I: CEVA_TDI
10 Reserved
11 Reserved

8000_2620h GPIO mode control register 3 GPIO_MODE3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO23_M		GPIO22_M		GPIO21_M		GPIO20_M		GPIO19_M		GPIO18_M		GPIO17_M		GPIO16_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO16_M GPIO mode selection
00 Configured as GPIO function
01 CEVA_TDO
10 Reserved
11 Reserved
GPIO17_M GPIO mode selection
00 Configured as GPIO function
01 O: CMOS Sensor Reset control Signal (CMRST)
10 O: DSP2_GPO3
11 O: Master DSP Task ID Bit 0 (D1_TID0)
GPIO18_M GPIO mode selection
00 Configured as GPIO function
01 O: CMOS Sensor Power Down control Signal (CMPDN)
10 O: DSP2_GPO2
11 O: Master DSP Task ID Bit 1 (D1_TID1)
GPIO19_M GPIO mode selection
00 Configured as GPIO function
01 I: CMVREF
10 O: DSP_GPO3
11 O: TDMA Debug (TBTXEN)
GPIO20_M GPIO mode selection
00 Configured as GPIO function
01 I: CMHREF
10 O: DSP_GPO2

11 O: TDMA Debug (TBTXFS)
GPIO21_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 8 (EINT8)
10 O: DSP_GPO1
11 O: TDMA Debug (TBRXEN)
GPIO22_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 9 (EINT9)
10 O: DSP_GPO0
11 O: TDMA Debug (TBRXFS)
GPIO23_M GPIO mode selection
00 Configured as GPIO function
01 Keyboard row 5 (KROW5)
10 O: DSP2_GPO1
11 Reserved

8000_2630h GPIO mode control register 4 GPIO_MODE4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31_M		GPIO30_M		GPIO29_M		GPIO28_M		GPIO27_M		GPIO26_M		GPIO25_M		GPIO24_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO24_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM6
10 O: DSP2_GPO0
11 Reserved
GPIO25_M GPIO mode selection
00 Configured as GPIO function
01 I2S_WS
10 I: Master DSP ICE CLK (D1ICK)
11 O: usb_probe_out[7]
GPIO26_M GPIO mode selection
00 Configured as GPIO function
01 I2S_CLK
10 Master DSP ICE DATA (D1ID)
11 O: usb_probe_out[6]
GPIO27_M GPIO mode selection
00 Configured as GPIO function
01 I2S_DAT
10 I: Master DSP ICE Mode Select (D1IMS)
11 O: usb_probe_out[5]

 © 2009 MediaTek Inc.

GPIO28_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: Keyboard column 5 (KCOL5)
- **10** I: Slave DSP ICE CLK (D2ICK)
- **11** O: usb_probe_out[4]

GPIO29_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Camera Strobe (CAM_STROBE)
- **10** Slave DSP ICE DATA (D2ID)
- **11** O: usb_probe_out[3]

GPIO30_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: CAM_MECHSH0 (Camera)
- **10** I: Slave DSP ICE Model Select (D2IMS)
- **11** O: usb_probe_out[2]

GPIO31_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: CAM_MECHSH1 (Camera)
- **10** O: Slave DSP Task ID Bit 0 (D2_TID0)
- **11** O: usb_probe_out[1]

8000_2640h GPIO mode control register 5 GPIO_MODE5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO39_M		GPIO38_M		GPIO37_M		GPIO36_M		GPIO35_M		GPIO34_M		GPIO33_M		GPIO32_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO32_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: usb_drvvbus
- **10** Reserved
- **11** Reserved

GPIO33_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: CMOS Sensor Flash control signal (CMFLASH)
- **10** O: Slave DSP Task ID Bit 2 (D2_TID2)
- **11** Reserved

GPIO34_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I2C Clock signal (SCL)
- **10** O: Slave DSP Task ID Bit 3 (D2_TID3)
- **11** Reserved

GPIO35_M GPIO mode selection

00 Configured as GPIO function
01 I2C Data Signal (SDA)
10 O: Slave DSP Task ID Bit 4 (D2_TID4)
11 Reserved

GPIO36_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM2
10 O: Slave DSP Task ID Bit 5 (D2_TID5)
11 Reserved

GPIO37_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM3
10 O: Slave DSP Task ID Bit 6 (D2_TID6)
11 CEVA_GPIO31

GPIO38_M GPIO mode selection
00 Configured as GPIO function
01 I: MC2WP
10 MC0DA4
11 Reserved

GPIO39_M GPIO mode selection
00 Configured as GPIO function
01 O: MC2PWRON
10 MC0DA5
11 Reserved

8000_2650h GPIO mode control register 6 GPIO_MODE6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47_M		GPIO46_M		GPIO45_M		GPIO44_M		GPIO43_M		GPIO42_M		GPIO41_M		GPIO40_M	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO40_M GPIO mode selection
00 Configured as GPIO function
01 MC2CM
10 MC0DA6
11 Reserved

GPIO41_M GPIO mode selection
00 Configured as GPIO function
01 I: MC2INS
10 MC0DA7
11 Reserved

GPIO42_M GPIO mode selection
00 Configured as GPIO function

01 O: Serial LCD Clock signal (LSCK)
10 O: TDMA Timer Debug Port Clock Output (TDMA_CK)
11 Reserved
GPIO43_M GPIO mode selection
00 Configured as GPIO function
01 O: Serial LCD Address signal (LSA0)
10 O: TDMA Timer Debug Port Data Output 1 (TDMA_D1)
11 O: TDMA Timer Debug (TDTIRQ)
GPIO44_M GPIO mode selection
00 Configured as GPIO function
01 Serial LCD Data signal (LSDA)
10 O: TDMA Timer Debug Port Data Output 0 (TDMA_D0)
11 O: TDMA Timer Debug (TCTIRQ2)
GPIO45_M GPIO mode selection
00 Configured as GPIO function
01 O: LSCE0B
10 O: TDMA Timer Debug Port Frame Sync signal (TDMA_FS)
11 O: TDMA Timer Debug (TCTIRQ1)
GPIO46_M GPIO mode selection
00 Configured as GPIO function
01 O: LSCE1B
10 O: LPCE2B
11 O: TDMA Timer Debug (TEVTVAL)
GPIO47_M GPIO mode selection
00 Configured as GPIO function
01 O: LPCE1B
10 O: D2_TID1
11 O: usb_probe_out[0]

8000_2660h GPIO mode control register 7 GPIO_MODE7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO55		GPIO54		GPIO53		GPIO52		GPIO51		GPIO50		GPIO49		GPIO48	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO48_M GPIO mode selection
00 Configured as GPIO function
01 I: JTRST_B
10 O: CLK_OUT6
11 Reserved
GPIO49_M GPIO mode selection
00 Configured as GPIO function
01 I: JTCK

10 O: CLK_OUT7
11 Reserved
GPIO50_M GPIO mode selection
00 Configured as GPIO function
01 I: JTDI
10 Reserved
11 Reserved
GPIO51_M GPIO mode selection
00 Configured as GPIO function
01 I: JTMS
10 Reserved
11 Reserved
GPIO52_M GPIO mode selection
00 Configured as GPIO function
01 JTDO
10 Reserved
11 Reserved
GPIO53_M GPIO mode selection
00 Configured as GPIO function
01 JRTCK
10 Reserved
11 Reserved
GPIO54_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM0
10 I: External interrupt input 16 (EINT16)
11 Reserved
GPIO55_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM1
10 I: BSI_RFIN
11 I: External interrupt input 17 (EINT17)

8000_2670h GPIO mode control register 8 GPIO_MODE8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63		GPIO62		GPIO61		GPIO60		GPIO59		GPIO58		GPIO57		GPIO56	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO56_M GPIO mode selection
00 Configured as GPIO function
01 O: VCXO enable output signal (SRCLKENA)
10 I: External interrupt input 18 (EINT18)

11 Reserved
GPIO57_M GPIO mode selection
00 Configured as GPIO function
01 O: VCXO enable output signal low active (SRCLKENAN)
10 I: External interrupt input 19 (EINT19)
11 Reserved
GPIO58_M GPIO mode selection
00 Configured as GPIO function
01 I: VCXO enable input signal (SRCLKENAI)
10 I: External interrupt input 20 (EINT20)
Note: This external interrupt only can be used by MD side, AP side can't use this interrupt.
11 Reserved
GPIO59_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 0 (EINT0)
10 Reserved
11 CEVA_GPIO18
GPIO60_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 1 (EINT1)
10 Reserved
11 CEVA_GPIO19
GPIO61_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 2 (EINT2)
10 Reserved
11 CEVA_GPIO20
GPIO62_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 3 (EINT3)
10 Reserved
11 CEVA_GPIO21
GPIO63_M GPIO mode selection
00 Configured as GPIO function
01 I: External interrupt input 4 (EINT4)
10 Reserved
11 CEVA_GPIO22

8000_2680h GPIO mode control register 9 GPIO_MODE9

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO71		GPIO70		GPIO69		GPIO68		GPIO67		GPIO66		GPIO65		GPIO64	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO64_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: External interrupt input 5 (EINT5)
- **10** Reserved
- **11** CEVA_GPIO23

GPIO65_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: External interrupt input 6 (EINT6)
- **10** Reserved
- **11** CEVA_GPIO24

GPIO66_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: External interrupt input 7 (EINT7)
- **10** Reserved
- **11** CEVA_GPIO25

GPIO67_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: URXD3
- **10** I: UART4 CTS signal (UCTS4)
- **11** Reserved

GPIO68_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: UTXD3
- **10** O: UART4 RTS signal (URTS4)
- **11** Reserved

GPIO69_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: URXD4
- **10** I: UART3 CTS signal (UCTS3)
- **11** Reserved

GPIO70_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: UTXD4
- **10** O: UART3 RTS signal (URTS3)
- **11** Reserved

GPIO71_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: Keyboard column 7 (KCOL7)
- **10** O: CLK_OUT4
- **11** CEVA_GPIO14

© 2009 MediaTek Inc.

8000_2690h GPIO mode control register 10 GPIO_MODE10

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO79		GPIO78		GPIO77		GPIO76		GPIO75		GPIO74		GPIO73		GPIO72	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO72_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: Keyboard column 6 (KCOL6)
- **10** O: CLK_OUT5
- **11** CEVA_GPIO15

GPIO73_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 7 (KROW7)
- **10** O: CLK_OUT6
- **11** CEVA_GPIO16

GPIO74_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 6 (KROW6)
- **10** O: CLK_OUT7
- **11** CEVA_GPIO17

GPIO75_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: Digital Audio Interface PCM Clock Output (DAICLK)
- **10** Reserved
- **11** CEVA_GPIO26

GPIO76_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: Digital Audio Interface PCM Data Output (DAIPCMOUT)
- **10** Reserved
- **11** CEVA_GPIO27

GPIO77_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: Digital Audio Interface PCM Data Input (DAIPCMIN)
- **10** Reserved
- **11** CEVA_GPIO28

GPIO78_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: Digital Audio Interface Reset Signal (DAIRST)
- **10** O: CLK_OUT5
- **11** CEVA_GPIO29

GPIO79_M GPIO mode selection

00 Configured as GPIO function
01 O: Digital Audio Interface Sync Signal (DAISYNC)
10 Reserved
11 CEVA_GPIO30

8000_26A0h GPIO mode control register 11 GPIO_MODE11

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO87		GPIO86		GPIO85		GPIO84		GPIO83		GPIO82		GPIO81		GPIO80	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO80_M GPIO mode selection
00 Configured as GPIO function
01 O: SPI_CS_N
10 I: IrDA RXD signal (IRDA_RXD)
11 O: BSI_CS1

GPIO81_M GPIO mode selection
00 Configured as GPIO function
01 O: SPI_SCK
10 O: IrDA TXD signal (IRDA_TXD)
11 Reserved

GPIO82_M GPIO mode selection
00 Configured as GPIO function
01 O: SPI_MOSI
10 O: IrDA power down control signal (IRDA_PDN)
11 MC2DA2

GPIO83_M GPIO mode selection
00 Configured as GPIO function
01 I: SPI_MISO
10 I: MIRQ
11 MC2DA3

GPIO84_M GPIO mode selection
00 Configured as GPIO function
01 I: IrDA RXD signal (IRDA_RXD)
10 CEVA_GPIO0
11 I: MFIQ

GPIO85_M GPIO mode selection
00 Configured as GPIO function
01 O: IrDA TXD signal (IRDA_TXD)
10 CEVA_GPIO1
11 Reserved

GPIO86_M GPIO mode selection
00 Configured as GPIO function

01 O: IrDA power down control signal (IRDA_PDN)
10 CEVA_GPIO2
11 Reserved

GPIO87_M GPIO mode selection
00 Configured as GPIO function
01 I: Keyboard column 0 (KCOL0)
10 CEVA_GPIO3
11 I: External interrupt input 10 (EINT10)

8000_26B0h GPIO mode control register 12 GPIO_MODE12

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO95		GPIO94		GPIO93		GPIO92		GPIO91		GPIO90		GPIO89		GPIO88	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO88_M GPIO mode selection
00 Configured as GPIO function
01 I: Keyboard column 1 (KCOL1)
10 CEVA_GPIO4
11 I: External interrupt input 11 (EINT11)

GPIO89_M GPIO mode selection
00 Configured as GPIO function
01 I: Keyboard column 2 (KCOL2)
10 CEVA_GPIO5
11 I: External interrupt input 12 (EINT12)

GPIO90_M GPIO mode selection
00 Configured as GPIO function
01 I: Keyboard column 3 (KCOL3)
10 CEVA_GPIO6
11 I: External interrupt input 13 (EINT13)

GPIO91_M GPIO mode selection
00 Configured as GPIO function
01 I: Keyboard column 4 (KCOL4)
10 CEVA_GPIO7
11 I: External interrupt input 14 (EINT14)

GPIO92_M GPIO mode selection
00 Configured as GPIO function
01 I: BTDMP_DIN1
10 CEVA_GPIO8
11 Reserved

GPIO93_M GPIO mode selection
00 Configured as GPIO function
01 BTDMP_FSP1

10 CEVA_GPIO9
11 O: PHY_CLK
GPIO94_M GPIO mode selection
00 Configured as GPIO function
01 BTDMP_CLK1
10 CEVA_GPIO10
11 O: LINE_STATE0
GPIO95_M GPIO mode selection
00 Configured as GPIO function
01 O: BTDMP_DOUT_2
10 CEVA_GPIO11
11 O: LINE_STATE1

8000_26C0h GPIO mode control register 13 GPIO_MODE13

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO103		GPIO102		GPIO101		GPIO100		GPIO99		GPIO98		GPIO97		GPIO96	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO96_M GPIO mode selection
00 Configured as GPIO function
01 BTDMP_FSP2
10 CEVA_GPIO12
11 O: TRACEDATA14
GPIO97_M GPIO mode selection
00 Configured as GPIO function
01 BTDMP_CLK2
10 CEVA_GPIO13
11 O: TRACEDATA15
GPIO98_M GPIO mode selection
00 Configured as GPIO function
01 CEVA_RTCK
10 Reserved
11 Reserved
GPIO99_M GPIO mode selection
00 Configured as GPIO function
01 O: PWM5
10 Reserved
11 Reserved
GPIO100_M GPIO mode selection
00 Configured as GPIO function
01 I: J2TRST_B
10 Reserved

11 O: TRACEDATA8

GPIO101_M GPIO mode selection

00 Configured as GPIO function

01 I: J2TCK

10 Reserved

11 O: TRACEDATA9

GPIO102_M GPIO mode selection

00 Configured as GPIO function

01 I: J2TDI

10 Reserved

11 O: TRACEDATA10

GPIO103_M GPIO mode selection

00 Configured as GPIO function

01 I: J2TMS

10 Reserved

11 O: TRACEDATA11

8000_26D0h GPIO mode control register 14 GPIO_MODE14

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO111		GPIO110		GPIO109		GPIO108		GPIO107		GPIO106		GPIO105		GPIO104	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO104_M GPIO mode selection

00 Configured as GPIO function

01 J2TDO

10 Reserved

11 O: TRACEDATA12

GPIO105_M GPIO mode selection

00 Configured as GPIO function

01 J2RTCK

10 Reserved

11 O: TRACEDATA13

GPIO106_M GPIO mode selection

00 Configured as GPIO function

01 Reserved

10 Reserved

11 Reserved

GPIO107_M GPIO mode selection

00 Configured as GPIO function

01 HDQ

10 Reserved

11 Reserved

GPIO108_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: URTS2
- **10** Reserved
- **11** Reserved

GPIO109_M GPIO mode selection
- **00** Configured as GPIO function
- **01** I: UCTS2
- **10** Reserved
- **11** Reserved

GPIO110_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 0 (KROW0)
- **10** Reserved
- **11** Reserved

GPIO111_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 1 (KROW1)
- **10** Reserved
- **11** Reserved

8000_26E0h GPIO mode control register 15 GPIO_MODE15

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO119		GPIO118		GPIO117		GPIO116		GPIO115		GPIO114		GPIO113		GPIO112	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO112_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 2 (KROW2)
- **10** Reserved
- **11** Reserved

GPIO113_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 3 (KROW3)
- **10** Reserved
- **11** Reserved

GPIO114_M GPIO mode selection
- **00** Configured as GPIO function
- **01** Keyboard row 4 (KROW4)
- **10** Reserved
- **11** Reserved

GPIO115_M GPIO mode selection

00 Configured as GPIO function
01 O: CLK_OUT0
10 Reserved
11 Reserved

GPIO116_M GPIO mode selection
00 Configured as GPIO function
01 O: CLK_OUT1
10 Reserved
11 Reserved

GPIO117_M GPIO mode selection
00 Configured as GPIO function
01 O: CLK_OUT2
10 Reserved
11 Reserved

GPIO118_M GPIO mode selection
00 Configured as GPIO function
01 O: CLK_OUT3
10 Reserved
11 Reserved

GPIO119_M GPIO mode selection
00 Configured as GPIO function
01 O: CLK_OUT4
10 Reserved
11 Reserved

8000_26F0h GPIO mode control register 16 GPIO_MODE16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO127		GPIO126		GPIO125		GPIO124		GPIO123		GPIO122		GPIO121		GPIO120	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO120_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO121_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO122_M GPIO mode selection
00 Configured as GPIO function

01 Reserved
10 Reserved
11 Reserved

GPIO123_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO124_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO125_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO126_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO127_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

8000_2700h GPIO mode control register 17 GPIO_MODE17

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO135		GPIO134		GPIO133		GPIO132		GPIO131		GPIO130		GPIO129		GPIO128	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO128_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO129_M GPIO mode selection
00 Configured as GPIO function
01 Reserved

10 Reserved
11 I: UCTS4

GPIO130_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 O: URTS4

GPIO131_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 I: UCTS3

GPIO132_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 O: URTS3

GPIO133_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO134_M GPIO mode selection
00 Configured as GPIO function
01 Reserved
10 Reserved
11 Reserved

GPIO135_M GPIO mode selection
00 Configured as GPIO function
01 SCL2
10 Reserved
11 Reserved

8000_2710h GPIO mode control register 18 GPIO_MODE18

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO143		GPIO142		GPIO141		GPIO140		GPIO139		GPIO138		GPIO137		GPIO136	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	01		01		01		01		01		01		01		01	

GPIO136_M GPIO mode selection
00 Configured as GPIO function
01 SDA2
10 Reserved

11 Reserved

GPIO137_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACECLK

10 Reserved

11 Reserved

GPIO138_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACECTL

10 Reserved

11 Reserved

GPIO139_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACEDATA0

10 Reserved

11 Reserved

GPIO140_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACEDATA1

10 Reserved

11 Reserved

GPIO141_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACEDATA2

10 Reserved

11 Reserved

GPIO142_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACEDATA3

10 Reserved

11 Reserved

GPIO143_M GPIO mode selection

00 Configured as GPIO function

01 O: TRACEDATA4

10 Reserved

11 Reserved

8000_2720h GPIO mode control register 19 GPIO_MODE19

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											GPIO146		GPIO145		GPIO144	
Type											R/W		R/W		R/W	
Reset											01		01		01	

GPIO144_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: TRACEDATA5
- **10** Reserved
- **11** Reserved

GPIO145_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: TRACEDATA6
- **10** Reserved
- **11** Reserved

GPIO146_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: TRACEDATA7
- **10** Reserved
- **11** Reserved

8000_2900h CLK_OUT0 setting CLK_OUT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT0			
Type													R/W			
Reset													0			

CLK_OUT0 select the clock output source of clk_out0
- **0** APMCUSYS bus clock (ahmclk_ck)
- **1** dsp1io_ck
- **2** f13m_ck
- **3** f65m_ck
- **4** f48m_ck
- **5** f32k_ck
- **6** f26m_ck
- **7** MDMCUSYS bus clock (mhmclk_ck)

8000_2910h CLK_OUT1 setting CLK_OUT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT1			
Type													R/W			
Reset													0			

CLK_OUT1 select the clock output source of clk_out1
- **0** APMCUSYS bus clock (ahmclk_ck)
- **1** dsp1io_ck
- **2** f13m_ck
- **3** f65m_ck
- **4** f48m_ck
- **5** f32k_ck

© 2009 MediaTek Inc.

6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2920h CLK_OUT2 setting CLK_OUT2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT2			
Type													R/W			
Reset													0			

CLK_OUT2 select the clock output source of clk_out2

0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck
2 f13m_ck
3 f65m_ck
4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2930h CLK_OUT3 setting CLK_OUT3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT3			
Type													R/W			
Reset													0			

CLK_OUT3 select the clock output source of clk_out3

0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck
2 f13m_ck
3 f65m_ck
4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2940h CLK_OUT4 setting CLK_OUT4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT4			
Type													R/W			
Reset													0			

CLK_OUT4 select the clock output source of clk_out4

0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck
2 f13m_ck
3 f65m_ck

4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2950h CLK_OUT5 setting CLK_OUT5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT5			
Type													R/W			
Reset													0			

CLK_OUT5 select the clock output source of clk_out5
0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck
2 f13m_ck
3 f65m_ck
4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2960h CLK_OUT6 setting CLK_OUT6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT6			
Type													R/W			
Reset													0			

CLK_OUT6 select the clock output source of clk_out6
0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck
2 f13m_ck
3 f65m_ck
4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

8000_2970h CLK_OUT7 setting CLK_OUT7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT7			
Type													R/W			
Reset													0			

CLK_OUT7 select the clock output source of clk_out7
0 APMCUSYS bus clock (ahmclk_ck)
1 dsp1io_ck

2 f13m_ck
3 f65m_ck
4 f48m_ck
5 f32k_ck
6 f26m_ck
7 MDMCUSYS bus clock (mhmclk_ck)

2.16 General Purpose Timer (AP)

2.16.1 General Description

Three general-purpose timers are provided. The timers are 16 bits long and run independently of each other, although they share the same clock source. Two timers can operate in one of two modes: one-shot mode and auto-repeat mode; the other is a free running timer. In one-shot mode, when the timer counts down and reaches zero, it is halted. In auto-repeat mode, when the timer reaches zero, it simply resets to countdown initial value and repeats the countdown to zero; this loop repeats until the disable signal is set to 1. Regardless of the timer's mode, if the countdown initial value (i.e. APGPT1_DAT for APGPT1 or APGPT_DAT2 for APGPT2) is written when the timer is running, the new initial value does not take effect until the next time the timer is restarted. In auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the general purpose timer, the desired values for APGPT_DAT and the APGPT_PRESCALER registers must first be set.

2.16.2 Register Definitions

APGPT+0000h GPT1 Control register APGPT1_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN**	**MODE**														
Type	R/W	R/W														
Reset	0	0														

MODE This register controls GPT1 to count repeatedly (in a loop) or just one-shot.

- **0** One-shot mode is selected.
- 1 Auto-repeat mode is selected.

EN This register controls GPT1 to start counting or to stop.

- **0** GPT1 is disabled.
- **1** GPT1 is enabled.

APGPT+0004h GPT1 Time-Out Interval register APGPT1_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT [15:0]**															
Type	R/W															
Reset	FFFFh															

CNT [15:0] Initial counting value. GPT1 counts down from GPT1_DAT. When GPT1 counts down to zero, a GPT1 interrupt is generated.

APGPT+0008h GPT2 Control register — APGPT2_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN**	**MODE**														
Type	R/W	R/W														
Reset	0	0														

MODE This register controls GPT2 to count repeatedly (in a loop) or just one-shot.

- **0** One-shot mode is selected
- **1** Auto-repeat mode is selected

EN This register controls GPT2 to start counting or to stop.

- **0** GPT2 is disabled.
- **1** GPT2 is enabled.

APGPT+000Ch GPT2 Time-Out Interval register — APGPT2_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT [15:0]**															
Type	R/W															
Reset	FFFFh															

CNT [15:0] Initial counting value. GPT2 counts down from GPT2_DAT. When GPT2 counts down to zero, a GPT2 interrupt is generated.

APGPT+0010h GPT Status register — APGPT_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**GPT2**	**GPT1**
Type															RC	RC
Reset															0	0

This register illustrates the gptimer timeout status. Each flag is set when the corresponding timer countdown completes, and can be cleared when the CPU reads the status register.

APGPT+0014h GPT1 Prescaler register — APGPT1_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer1.

- **000** 16384 Hz
- **001** 8192 Hz
- **010** 4096 Hz
- **011** 2048 Hz
- 100 1024 Hz
- 101 512 Hz

110 256 Hz
111 128 Hz

APGPT+0018h GPT2 Prescaler register APGPT2_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer2.

000 16384 Hz
001 8192 Hz
010 4096 Hz
011 2048 Hz
100 1024 Hz
101 512 Hz
110 256 Hz
111 128 Hz

APGPT+001Ch GPT3 Control register APGPT3_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**EN**
Type																R/W
Reset																0

EN This register controls GPT3 to start counting or to stop.

0 GPT3 is disabled.
1 GPT3 is enabled.

APGPT+0020h GPT3 Time-Out Interval register APGPT3_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT[15:0]**															
Type	RO															
Reset	0															

CNT [15:0] If EN=1, GPT3 is a free running timer . Software reads this register for the countdown start value for GPT3.

APGPT+0024h GPT3 Prescaler register APGPT3_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer3.

000 16384 Hz
001 8192 Hz
010 4096 Hz
011 2048 Hz
100 1024 Hz
101 512 Hz
110 256 Hz
111 128 Hz

2.17 GRAPH1SYS Clock Management Register

CONFG_BASE = 0x80092000

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CONFG_BASE + 300h	Clock Gating Control Status Register	GRAPH1SYS_CG_CON
CONFG_BASE + 320h	Clock Gating Set Register	GRAPH1SYS_CG_SET
CONFG_BASE + 340h	Clock Gating Clear Register	GRAPH1SYS_CG_CLR

Table 35 ***APB Bridge Register Map***

2.17.1 Register Definitions

CONFG_BASE + 300h **Clock Gating Control Status Register** **GRAPH1SYS_CG_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							RO	RO	RO	RO		RO	RO		RO	RO
Reset							1	1	1	1		1	1		1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGDMA0	BLS	GCMQ	G2D	GMC1
Type		RO	RO	RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO
Reset		1	1	1	1	1	1	1		1	1	1	1	1	1	0

GRAPH1 sub-system clock gating control status register (read only), value 1 represents clock gating.

GMC1 Status of the GMC1 Clock Gating.
G2D Status of the G2D Clock Gating.
GCMQ Status of the GCMQ Clock Gating.
BLS Status of the BLS Clock Gating.
IMGDMA0 Status of the IMGDMA0 Clock Gating.
PNG Status of the PNG Clock Gating.

DSI Status of the DSI Clock Gating.
TVE Status of the TVE Clock Gating.
TVC Status of the TVC Clock Gating.
ISP Status of the ISP Clock Gating.
IPP Status of the IPP Clock Gating.
PRZ Status of the PRZ Clock Gating.
CRZ Status of the CRZ Clock Gating.
DRZ Status of the DRZ Clock Gating.
WT Status of the WT Clock Gating.
AFE Status of the AFE Clock Gating.
SPI Status of the SPI Clock Gating.
ASM Status of the ASM Clock Gating.
RESZ_LB Status of the RESZ_LB Clock Gating.
LCD Status of the LCD Clock Gating.
DPI Status of the DPI Clock Gating.
G1FAKE Status of the G1FAKE Clock Gating.

CONFG_BASE + 320h Clock Gating Set Register GRAPH1SYS_CG_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							WO	WO	WO	WO		WO	WO		WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGDMA0	BLS	GCMQ	G2D	GMC1
Type		WO	WO	WO	WO	WO	WO	WO		WO	WO	WO	WO	WO	WO	WO

GRAPH1 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_COND registers.
Eg.
If PDN_COND = 16’h0F0F,
Writing PDN_COND = 16’F0F0 will result in PDN_COND = 16’hFFFF.

GMC1 Set GMC1 Clock Gating.
G2D Set G2D Clock Gating.
GCMQ Set GCMQ Clock Gating.
BLS Set BLS Clock Gating.
IMGDMA0 Set IMGDMA0 Clock Gating.
PNG Set PNG Clock Gating.
DSI Set DSI Clock Gating.
TVE Set TVE Clock Gating.
TVC Set TVC Clock Gating.
ISP Set ISP Clock Gating.

IPP Set IPP Clock Gating.
PRZ Set PRZ Clock Gating.
CRZ Set CRZ Clock Gating.
DRZ Set DRZ Clock Gating.
WT Set WT Clock Gating.
AFE Set AFE Clock Gating.
SPI Set SPI Clock Gating.
ASM Set ASM Clock Gating.
RESZ_LB Set RESZ_LB Clock Gating.
LCD Set LCD Clock Gating.
DPI Set DPI Clock Gating.
G1FAKE Set G1FAKE Clock Gating.

CONFG_BASE + 340h Clock Gating Clear Register GRAPH1SYS_CG_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							WO	WO	WO	WO		WO	WO		WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGDMA0	BLS	GCMQ	G2D	GMC1
Type		WO	WO	WO	WO	WO	WO	WO		WO	WO	WO	WO	WO	WO	WO

GRAPH1 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_COND registers.
Eg.
If PDN_COND = 16’hFFFF,
Writing PDN_COND = 16’F0F0 will result in PDN_COND = 16’h0F0F.

GMC1 Clear GMC1 Clock Gating.
G2D Clear G2D Clock Gating.
GCMQ Clear GCMQ Clock Gating.
BLS Clear BLS Clock Gating.
IMGDMA0 Clear IMGDMA0 Clock Gating.
PNG Clear PNG Clock Gating.
DSI Clear DSI Clock Gating.
TVE Clear TVE Clock Gating.
TVC Clear TVC Clock Gating.
ISP Clear ISP Clock Gating.
IPP Clear IPP Clock Gating.
PRZ Clear PRZ Clock Gating.
CRZ Clear CRZ Clock Gating.

DRZ	Clear DRZ Clock Gating.
WT	Clear WT Clock Gating.
AFE	Clear AFE Clock Gating.
SPI	Clear SPI Clock Gating.
ASM	Clear ASM Clock Gating.
RESZ_LB	Clear RESZ_LB Clock Gating.
LCD	Clear LCD Clock Gating.
DPI	Clear DPI Clock Gating.
G1FAKE	Clear G1FAKE Clock Gating.

2.18 GRAPH2SYS Clock Management Register

CONFG_BASE = 0x800A7000

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CONFG_BASE + 000h	Clock Gating Control Status Register	GRAPH2SYS_CG_CON
CONFG_BASE + 004h	Clock Gating Set Register	GRAPH2SYS_CG_SET
CONFG_BASE + 008h	Clock Gating Clear Register	GRAPH2SYS_CG_CLR
CONFG_BASE + 010h	Memory Delsel Control Regsiter 0	GRAPH2SYS_DELSEL0
CONFG_BASE + 014h	Memory Delsel Control Regsiter 1	GRAPH2SYS_DELSEL1
CONFG_BASE + 018h	Memory Delsel Control Regsiter 2	GRAPH2SYS_DELSEL2

Table 36 ***APB Bridge Register Map***

2.18.1 Register Definitions

CONFG_BASE + 000h **Clock Gating Control Status Register** **GRAPH2SYS_CG_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset								1	1	1	1	1	1	1	1	0

GRAPH2 sub-system clock gating control status register (read only), value 1 represents clock gating.

GMC2 Status of the GMC2 Clock Gating.

IMAGE_DMA_1 Status of the IMAGE_DMA_1 Clock Gating.
PRZ Status of the PRZ Clock Gating.
M3D Status of the M3D Clock Gating.
H264 Status of the H264 Clock Gating.
DCT Status of the DCT Clock Gating.
JPEG Status of the JPEG Clock Gating.
MP4 Status of the MP4 Clock Gating.
MP4_DEBLK Status of the MP4_DEBLK Clock Gating.

CONFG_BASE + 004h Clock Gating Set Register GRAPH2SYS_CG_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2
Type								W/O	WO	WO	WO	WO	WO	WO	WO	WO

GRAPH2 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.
Eg.
If CG_CON = 16’h0F0F,
Writing CG_SET = 16’F0F0 will result in CG_CON = 16’hFFFF.

GCU Set the GCU Controller Power Down.
GMC2 Set the GMC2 Clock Gating.
IMAGE_DMA_1 Set the IMAGE_DMA_1 Clock Gating.
PRZ Set the PRZ Clock Gating.
M3D Set the M3D Clock Gating.
H264 Set the H264 Clock Gating.
DCT Set the DCT Clock Gating.
JPEG Set the JPEG Clock Gating.
MP4 Set the MP4 Clock Gating.
MP4_DEBLK Set the MP4_DEBLK Clock Gating.

CONFG_BASE + 008h Clock Gating Clear Register GRAPH2SYS_CG_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2

Type								WO	WO	WO	WO	WO	WO	WO	WO	WO

GRAPH2 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding "Clear" register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.
Eg.
If CG_CON = 16'hFFFF,
Writing CG_CLR = 16'F0F0 will result in CG_CON = 16'h0F0F.

GCU Clear the GCU Controller Power Down.
GMC2 Clear the GMC2 Clock Gating.
IMAGE_DMA_ Clear the IMAGE_DMA_1 Clock Gating.
PRZ Clear the PRZ Clock Gating.
M3D Clear the M3D Clock Gating.
H264 Clear the H264 Clock Gating.
DCT Clear the DCT Clock Gating.
JPEG Clear the JPEG Clock Gating.
MP4 Clear the MP4 Clock Gating.
MP4_DEBLK Clear the MP4_DEBLK Clock Gating.

2.19 HDQ/1-wire

2.19.1 General Description

The HDQ/1-wire design comprises the following blocks:

- HDQ finite state machine (HDQ_FSM)
- HDQ register (HDQ_REG)
- HDQ interrupt controller (HDQ_INT)

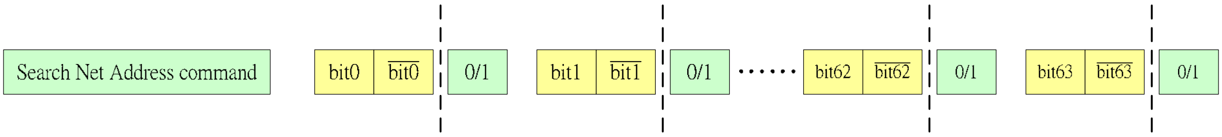
HDQ is a point to point communication. It uses a single wire to communication between master and slave. The protocol is an asynchronous return-to-one mechanism referenced to Vss. The protocol is byte access between master and slave. In a typical write to slave (or read from slave), one byte command will be sent by master, and then another byte data will be sent to slave (or be read from slave). Some slave also support for two bytes access which can be wrote or read 16 bits data with a master command. HDQ pin is an open-drain device. It means that the HDQ bus need an external pull-up resistance to Vss.
1-wire interface is a point to multi-point communication. It is also a single wire to communication between master and slave. And it also an asynchronous return-to-one mechanism referenced to Vss. Each slave has a 64 bit net address as follow:

8-bit CRC	48-bit serial number	8-bit family code
MSB		LSB

In additional to Search Net Address command, each 1-wire communication is formed by 1) one byte command only, 2) one byte command with 16 bit data read/write, 3) one byte command with 8 bit data read/write, or 4) one byte command with 64 bit net address read/write.

command	+	Read/write data

When the master send a Search Net Address command, the slave might return two bit, and then the master shall return one bit to select the device. SW can just read two bits data from slave by setting 0x04[11:10]. And we can return the select bit as a one bit command by setting 0x04[9].



2.19.2 Address Map for HDQ/1-wire

REGISTER ADDRESS	REGISTER NAME
Common Control Registers	
HDQ + 0000h	HDQ/1-wire tx command
HDQ + 0004h	HDQ/1-wire control
HDQ + 0008h	1-wire control
HDQ + 000Ch	HDQ div contol
HDQ + 0010h	1-wire status (read-only)
HDQ + 0014h	HDQ/1-wire interrupt enable
HDQ + 0018h	HDQ/1-wire interrupt clear (write-only)
HDQ + 001Ch	HDQ/1-wire interrupt (read-only)
HDQ + 0020h ~ 002ch	HDQ/1-wire data write
HDQ + 0030h ~ 003ch	HDQ/1-wire data read (read-only)
HDQ + 0040h ~ 004ch	HDQ timing parameter
HDQ + 0050h ~ 0064h	1-wire timing parameter

Blue for R/W,

Yellow for Read only,

Gyay for reserved.

Green for Write only

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	tx_command[7:0]							
4		sw_rst	no_command	wr_rd_data_type[2:0]			one_bit_command	Overdrive
5	hdq_1-wire_en	hdq_1-wire_wr	rate_fix[1:0]		gobit	hdq16_en	break	hdq_en
8	onewire_sample_time[15:8]							
C	div[7:0]							
10					rx_always_high	1-wire_presence_short	1-wire_presence_long	rx_always_low
14		1-wire_alarm_en	1-wire_presence_en	1-wire_presence_timeout_en	hdq_rx_timeout_en	hdq_break_intr_en	hdq_rx_done_en	hdq_tx_done_en
18		1-wire_alarm_clr	1-wire_presence_clr	1-wire_presence_timeout_clr	hdq_rx_timeout_clr	hdq_break_intr_clr	hdq_rx_done_clr	hdq_tx_done_clr
1C		1-wire_alarm_intr	1-wire_presence_intr	1-wire_presence_timeout_intr	hdq_rx_timeout_intr	hdq_break_intr	hdq_rx_done_intr	hdq_tx_done_intr

© 2009 MediaTek Inc.

20~2F	hdq_tx_data[63:0]
30~3F	hdq_rx_data[63:0]
40	hdq_tb[7:0]
41	hdq_tbreak[7:0]
44	hdq_cych[7:0]
45	hdq_tsp[7:0]
48	hdq_thw0[7:0]
49	hdq_thw1[7:0]
4c	hdq_rxto[7:0]
50	onewire_tpdl_min[7:0]
51	onewire_tpdl_max[7:0]
54	onewire_tint1_min[7:0]
55	onewire_tint1_max[7:0]
58	onewire_tpdh [7:0]
59	onewire_trstl [7:0]
5c	onewire_tmslot [7:0]
5d	onewire_tsslot_max [7:0]
60	onewire_tlow0 [7:0]
61	onewire_tlow1 [7:0]
64	onewire_tdri [7:0]
65	onewire_trsth [7:0]

2.19.3 Register Definitions

HDQ+0000h HDQ/1-wire tx command HDQ_COM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									HDQ_TX_COMMAND							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0

HDQ_TX_COMMAND[7:0] 8-bits tx command. It will be sent when the gobit (0x04[11]) is set. In the Search Net Address sequence, SW might send 1-bit response data into hdq_tx_comand to select slave device.

HDQ+0004h HDQ/1-wire control HDQ_CTRL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		SW_RST	NO_COMMAND	DATA_TYPE			1BIT_COMMAND	OVERDRIVE	HDQ_1WIRE_EN	HDQ_1WIRE_TX	RATE_FIX		GOBIT	HDQ16_EN	BREAK	HDQ_EN
Type		R/w	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	WO	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

SW_RST Software can reset hardware state machine and interrupt by setting this bit to 1. APB registers in

this document will not be reset by SW_RST. Remember to set this bit to 0 after finishing SW reset.
NO_COMMAND It will be allowed that the data will be sent or received without sent any command in front of it if this bit is set. 1bit_command (0x04[9]) will be ignored when this bit is set, and data_type (0x04[12:10]) just can be 0, 1, or 3 when this bit is set. This bit is only used for 1-wire bus (0x04[0]=0).
DATA_TYPE[2:0] The type of the wr/rd data following after command. It only used for 1-wire bus (0x04[0]=0).
3'b000: 2 bytes data.
3'b001: 8 bytes data
3'b010: 2 bit data
3'b011: 1 byte data
3'b100: no data is following
Others are reserved to define.
1 BIT COMMAND This bit is used only when enabling 1-wire bus (0x04[0]=0). When this bit is set to 0, 8-bit tx command in 0x00 will be sent when gobit (0x04[3]) is set to 1'b1. Otherwire, only 0x00[0] will be sent when gobit is set. The 1-bit command is used for Search Net Address sequence.
OVERDRIVE This bit is used only when enabling 1-wire bus (0x04[0]=0). When this bit is set to 0, the 1-wire bus is in the standard mode. Otherwise, the 1-wire bus is in overdrive mode.
HDQ_1WIRE_EN This bit is used to enable the HDQ/1-wire bus function. When this bit is set to 1'b0, the HDQ/1-wire will be disabled.
HDQ_1WIRE_TX When this bit is set to 1'b1, it means that the data following the command will be writen into the slave. Otherwire, the following data is read from slave.
RATE_FIX[1:0] When rate_fix is set to 2'b10, it means that the device is operated when f13m_en is enable. When rate_fix is set to 2'b11, it means that the device is operated when f26m_en is enable. Otherwire, the device is operated with the input clock. When the device is operated for 1-wire bus, the clock into the design shall be 13MHz, or the rate_fix shall be set to 13MHz.
GOBIT When this bit is set to 1'b1, the command will be sent and then the data might be sent or received if the BREAK bit (0x04[1]) is set to 0. If the BREAK is set to 1'b1 and we setting the GOBIT, 1) a BREAK signal will be sent when HDQ_EN (0x04[0]) is set to 1, or 2) a RESET signal will be sent from master and the slave shall response a PRESENCE signal to master. This GOBIT is write-only and it will auto be reset after the signal is sent.
HDQ16_EN This bit is only used for HDQ bus (0x04[0]=1). When this bit is set to 1'b1, the wr/rd data following the command is 16-bit. Otherwise, the data will be 8-bit.
BREAK When the device is used for HDQ bus (0x04[0]=1), a BREAK signal will be sent when this bit is set to 1 and the GOBIT (0x04[3]) is set to 1. When the device is used for 1-wire bus (0x04[0]=0), a RESET signal will be sent from master and the slave shall response a PRESENCE signal to master when this bit is set to 1 and the GOBIT (0x04[3]) is set to 1.
HDQ_EN When this bit is set to 1'b1, it means that device is used for HDQ bus. Otherwise, the device is used for 1-wire bus. When you use this as a 1-wire device, a pull-up resistance must be added in the hdq inout port.

HDQ+0008h 1-wire control 1WIRE_CTRL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	1WIRE_SAMPLE_TIME[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

1WIRE_SAMPLE_TIME[15:0] This register is only used for 1-wire bus. This value is the sample time after the negtive edge in the bus when data is read from slave. Each step is about 76.92ns (1/13MHz).

HDQ+000Ch HDQ DIV Register HDQ_DIV

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									HDQ_DIV[7:0]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset									0	0	0	0	1	0	1	1

HDQ_DIV[7:0] This register is only used for HDQ bus. When the clock input into design is 52MHz, this value shall set to 52. If the clock is 26MHz or the rate_fix is set to operate. for 26MHz, this value shall set to 26. If the clock is 13MHz or the rate_fix is set to operated for 13MHz, this value shall set to 13.

HDQ+0010h 1WIRE Status Register 1WIRE_STAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RX_ALWAYS_HIGH	1WIRE_PRESENCE_SHORT	1WIRE_PRESENCE_LONG	RX_ALWAYS_LOW
Type													RO	RO	RO	RO
Reset													0	0	0	0

RX_ALWAYS_HIGH When this bit is set to 1, it means that the device the bus is always high even if we send 0 to the bus.
1WIRE_PRESENCE_SHORT When this bit is set to 1, it means that the presence signal from slave is shorter than we expect.
1WIRE_PRESENCE_LONG When this bit is set to 1, it means that the presence signal from slave is longer than we expect.
RX_ALWAYS_LOW When this bit is set to 1, it means that the device the bus is always low when we don't drive the bus.

HDQ+0014h HDQ/1-wire INTR Enable HDQ_INT_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										1WIRE_ALARM_EN	1WIRE_PRESENCE_EN	1WIRE_PRESENCE_TIMEOUT_EN	HDQ_RX_TIMEOUT_EN	HDQ_BREAK_EN	HDQ_RX_DONE_EN	HDQ_TX_DONE_EN
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

1WIRE_ALARM_EN When this bit is set to 1, the alarm interrupt will be occurred if an alarm signal is detected. It is only used for 1-wire bus.
1WIRE_PRESENCE_EN When this bit is set to 1, the presence interrupt will be occurred if a presence signal is detected. It is only used for 1-wire bus.
1WIRE_PRESENCE_TIMEOUT_EN When this bit is set to 1, the presence timeout interrupt will be occurred if no presence signal is received from slave after sending the RESET signal. It is only used for 1-wire bus.
HDQ_RX_TIMEOUT_EN When this bit is set to 1, the HDQ rx timeout interrupt will be occurred if the slave does not response anything after we send a read command. It is only used for HDQ bus.
HDQ_BREAK_EN When this bit is set to 1, the HDQ break interrupt will be occurred after the master send the break signal to slave. It is only used for HDQ bus.
HDQ_RX_DONE_EN When this bit is set to 1, the HDQ/1-wire rx done interrupt will be occurred after the master send the read command to slave and then receive the data from slave. It is used for both HDQ and 1-wire bus.
HDQ_TX_DONE_EN When this bit is set to 1, the HDQ/1-wire tx done interrupt will be occurred after the master send the read command and the data to the slave. It is used for both HDQ and 1-wire bus.

HDQ+0018h HDQ/1-wire INTR Clear HDQ_INT_CLR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name										1WIRE_ALARM_CLR	1WIRE_PRESENCE_CLR	1WIRE_PRESENCE_TIMEOUT_CLR	HDQ_RX_TIMEOUT_CLR	HDQ_BREAK_CLR	HDQ_RX_DONE_CLR	HDQ_TX_DONE_CLR
Type										WO	WO	WO	WO	WO	WO	WO
Reset										0	0	0	0	0	0	0

1WIRE_ALARM_CLR When this bit is set to 1, the alarm interrupt will be cleared. It is only used for 1-wire bus. And it will auto be reset by HW.
1WIRE_PRESENCE_CLR When this bit is set to 1, the presence interrupt will be cleared. It is only used for 1-wire bus. And it will auto be reset by HW.
1WIRE_PRESENCE_TIMEOUT_CLR When this bit is set to 1, the presence timeout interrupt will be cleared. It is only used for 1-wire bus. And it will auto be reset by HW.
HDQ_RX_TIMEOUT_CLR When this bit is set to 1, the HDQ rx timeout interrupt will be cleared. It is only used for HDQ bus. And it will auto be reset by HW.
HDQ_BREAK_CLR When this bit is set to 1, the HDQ break interrupt will be cleared. And it will auto be reset by HW.
HDQ_RX_DONE_CLR When this bit is set to 1, the HDQ/1-wire rx done interrupt will be cleared. It is used for both HDQ and 1-wire bus. And it will auto be reset by HW.
HDQ_TX_DONE_CLR When this bit is set to 1, the HDQ/1-wire tx done interrupt will be cleared. It is used for both HDQ and 1-wire bus. And it will auto be reset by HW.

HDQ+001Ch HDQ/1-wire INTR HDQ_INTR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										1WIRE_ALARM_INTR	1WIRE_PRESENCE_INTR	1WIRE_PRESENCE_TIMEOUT_INTR	HDQ_RX_TIMEOUT_INTR	HDQ_BREAK_INTR	HDQ_RX_DONE_INTR	HDQ_TX_DONE_INTR
Type										RO	RO	RO	RO	RO	RO	RO
Reset										0	0	0	0	0	0	0

1WIRE_ALARM_INTR This is the slave alarm interrupt. When this bit is set to 1, SW shall read the status register of the slave to check the alarm item. It is only used for 1-wire bus.
1WIRE_PRESENCE_INTR When this bit is set to 1, it means that the master had sent a RESET signal and then received a PRESENCE signal from slave. It is only used for 1-wire bus.
1WIRE_PRESENCE_TIMEOUT_INTR When this bit is set to 1, it means that the master had sent a RESET signal, but no PRESENCE is detected from the bus. It is only used for 1-wire bus.
HDQ_RX_TIMEOUT_INTR When the interrupt is occurred, it means that the the master had sent a read command, but no data is responded from slave. It is only used for HDQ bus.
HDQ_BREAK_INTR When this bit is set to 1, it means that the master had sent a BREAK signal to the slave. It is only used for HDQ bus.
HDQ_RX_DONE_INTR When this bit is set to 1, it means that the HDQ/1-wire master had sent a read command to the slave, and then received the data sent by slave. It is used for both HDQ and 1-wire bus.
HDQ_TX_DONE_INTR When this bit is set to 1, it means that the HDQ/1-wire master had sent a write command and data to the slave. It is used for both HDQ and 1-wire bus.

HDQ +0020h HDQ TX DATA HDQ_TX_DATA

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Name	**HDQ_TX_DATA[63:48]**															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Name	**HDQ_TX_DATA[47:32]**															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HDQ_TX_DATA[31:16]**															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**HDQ_TX_DATA[15:0]**															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDQ_TX_DATA[63:0] 64 bit tx data. The LSB will be sent first. For example, if the tx data is 16 bit, HDQ_TX_DATA[15:0] will be sent to the slave.

HDQ +0030h HDQ RX DATA HDQ_RX_DATA

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Name	**HDQ_RX_DATA[63:48]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Name	**HDQ_RX_DATA[47:32]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HDQ_RX_DATA[31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**HDQ_RX_DATA[15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDQ_RX_DATA[63:0] 64 bit rx data from slave. The LSB will be received first. For example, if the rx data is 16 bit, the receiving data from slave will be put in HDQ_RX_DATA[15:0].

HDQ+0040h HDQ PARA_1 HDQ_PARA_1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**HDQ_TBREAK[7:0]**								**HDQ_TB[7:0]**							
Type	R/W								R/W							
Reset	8'd230								8'd190							

HDQ_TBREAK[7:0] This register is defined the total HDQ break time as the diagram in 1.4.1. For example, if the value is 8'd230 as default, the time in HDQ break will be 230 us.

HDQ_TB[7:0] This register is defined the HDQ break low time as the diagram in 1.4.1. For example, if the value is 8'd190 as default, the low time in HDQ break will be 190 us.

HDQ+0044h HDQ PARA_2 HDQ_PARA_2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	HDQ_TSP[7:0]	HDQ_CYCH[7:0]
Type	R/W	R/W
Reset	8'd65	8'd190

HDQ_TSP[7:0] This register is defined the HDQ sample point as the diagram in 1.4.1. For example, if the value is 8'd65 as default, the HDQ sample point will be 65 us.

HDQ_CYCH[7:0] This register is defined the HDQ TX bit cycle time as the diagram in 1.4.1. For example, if the value is 8'd190 as default, the TX bit cycle time in HDQ will be 230 us.

HDQ+0048h HDQ PARA_3 HDQ_PARA_3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDQ_THW1[7:0]								HDQ_THW0[7:0]							
Type	R/W								R/W							
Reset	8'd40								8'd130							

HDQ_THW1[7:0] This register is defined the low duration on the bus when HDQ send a "1" to slave as the diagram in 1.4.1. For example, if the value is 8'd400 as default, the HDQ sample point will be 40 us.

HDQ_THW0[7:0] This register is defined the low duration on the bus when HDQ send a "0" to slave as the diagram in 1.4.1. For example, if the value is 8'd130 as default, the HDQ sample point will be 130 us.

HDQ+004Ch HDQ PARA_4 HDQ_PARA_4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									HDQ_RXTO[7:0]							
Type									R/W							
Reset									8'd130							

HDQ_RXTO[7:0] This register is defined the HDQ RX timeout value. If the HDQ bus is not driven to 0 (by slave) for more than this setting value, the HDQ RX timeout interrupt will be set.
The timeout time will be (2 * HDQ_RXTO) us. For example, if this register is 8'd130 as default, the HDQ RX timeout time will be 130*2=260us.

HDQ+0050h 1-WIRE PARA_1 1WIRE_PARA_1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONEWIRE_TPDL_MAX[7:0]								ONEWIRE_TPDL_MIN[7:0]							
Type	R/W								R/W							
Reset	8'd195								8'd195							

onewire_Tpdl_max[7:0] This register is defined the 1-wire bus presense max. time as the diagram in 1.4.2.
The max. presense time will be (16 * onewire_tpdl_min * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the max. presense time will be (16*195*0.07692)=240us. Be careful, (onewire_tint1_max*256) need to be larger than (onewire_tpdl_max*16)!

onewire_Tpdl_min[7:0] This register is defined the 1-wire bus presense min. time as the diagram in 1.4.2.
The min. presense time will be (4 * onewire_tpdl_min * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the min. presense time will be (4*195*0.07692)=60us. Be careful, (onewire_trsth*32) need to be larger than (onewire_tpdl_min*4).

HDQ+0054h 1-WIRE PARA_2 1WIRE_PARA_2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONEWIRE_TINT1_MAX[7:0]								ONEWIRE_TINT1_MIN[7:0]							
Type	R/W								R/W							
Reset	8'd220								8'd195							

 © 2009 MediaTek Inc.

onewire_Tint1_max[7:0] This register is defined the 1-wire bus max. interrupt duration time as the diagram in 1.4.3. The max. interrupt duration time will be (256 * onewire_tint1_max * input_clock_period). For example, if this register is 8'd220 as default and the input clock is 13MHz (76.92ns), the max. interrupt duration time will be (256*220*0.07692)=4332 us. Be careful, (onewire_tint1_max*256) need to be larger than (onewire_tpdl_max*16)!

onewire_Tint1_min[7:0] This register is defined the 1-wire bus min. interrupt duration time as the diagram in 1.4.3. The min. interrupt duration time will be (64 * onewire_tint1_min * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the min. interrupt duration time will be (64*195*0.07692)=960 us. Be careful, (onewire_tint1_min*64) need to be larger than (onewire_trstl*32)!

HDQ+0058h 1-WIRE PARA_3 1WIRE_PARA_3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONEWIRE_TRTSL[7:0]								ONEWIRE_TPDH[7:0]							
Type	R/W								R/W							
Reset	8'd203								8'd195							

onewire_Trtsl[7:0] This register is defined the 1-wire bus reset low duration time as the diagram in 1.4.2. The reset low duration time will be (32 * onewire_trtsl * input_clock_period). For example, if this register is 8'd203 as default and the input clock is 13MHz (76.92ns), the 1-wire bus reset low duration time will be (32*203*0.07692)=500 us. Be careful, (onewire_tint1_min*64) need to be larger than (onewire_trstl*32)!

onewire_Tpdh[7:0] This register is defined the duration time from reset to presense as the diagram in 1.4.2. The the duration time from reset to presense will be (4 * onewire_tpdh * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the duration time from reset to presense will be (4*195*0.07692)=60 us.

HDQ+005Ch 1-WIRE PARA_4 1WIRE_PARA_4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONEWIRE_TSSLOT_MAX[7:0]								ONEWIRE_TMSLOT[7:0]							
Type	R/W								R/W							
Reset	8'd195								8'd163							

onewire_Tsslot_max[7:0] This register is defined the 1-wire bus slave tx max. bit duration as the diagram in 1.4.2. The slave tx max bit duration time will be (8 * onewire_tsslot_max * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the slave tx max bit duration time will be (8*195*0.07692)=120 us.

onewire_Tmslot[7:0] This register is defined the 1-wire bus master tx bit duration as the diagram in 1.4.2. The master tx bit duration time from reset to presense will be (8 * onewire_tmslot * input_clock_period). For example, if this register is 8'd163 as default and the input clock is 13MHz (76.92ns), the min. presense time will be (8*163*0.07692)=100 us.

HDQ+0060h 1-WIRE PARA_5 1WIRE_PARA_5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONEWIRE_TLOW1[7:0]								ONEWIRE_TLOW0[7:0]							
Type	R/W								R/W							
Reset	8'd170								8'd130							

onewire_Tlow1[7:0] This register is defined the bus driven low duration when the master send a bit "1" to slave, as the diagram in 1.4.2. The bus driven low duration time when the master send a bit "1" to slave will be (1 * onewire_tlow1 * input_clock_period). For example, if this register is 8'd170 as default and the input clock is 13MHz (76.92ns), the slave tx max bit duration time will be (1*170*0.07692)=13 us.

onewire_Tlow0[7:0] This register is defined the bus driven low duration when the master send a bit "0" to slave, as the diagram in 1.4.2. The bus driven low duration time when the master send a bit "0" to slave will

be (8 * onewire_tlow0 * input_clock_period). For example, if this register is 8'd130 as default and the input clock is 13MHz (76.92ns), the slave tx max bit duration time will be (8*130*0.07692)=80 us.

HDQ+0064h 1-WIRE PARA_6 1WIRE_PARA_6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ONEWIRE_TRSTH[7:0]**								**ONEWIRE_TDRI[7:0]**							
Type	R/W								R/W							
Reset	8'd195								8'd17							

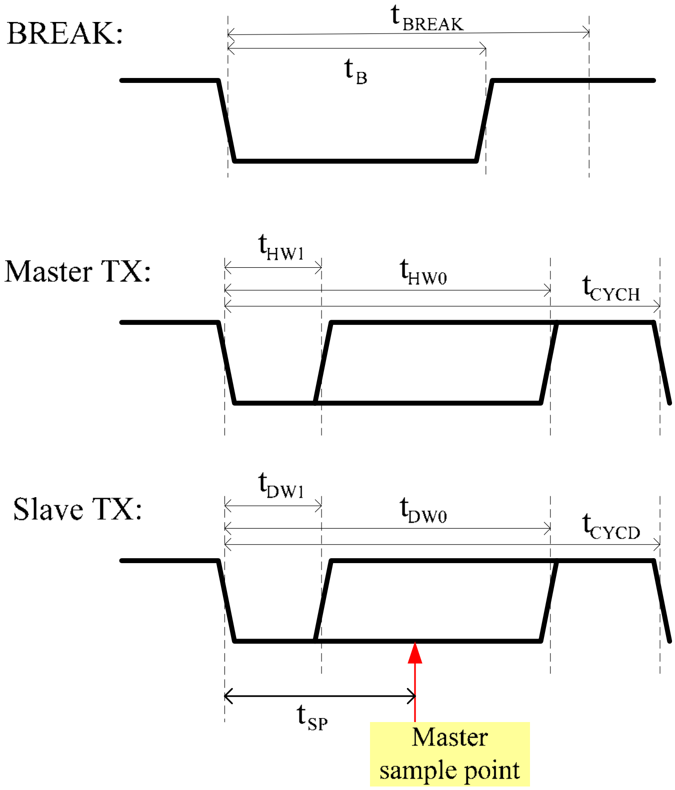
onewire_Trsth[7:0] This register is defined the presense process duration after master send the reset signal, as the diagram in 1.4.2. This duration will be (32 * onewire_trsth * input_clock_period). For example, if this register is 8'd195 as default and the input clock is 13MHz (76.92ns), the slave tx max bit duration time will be (32*195*0.07692)=480 us. Be careful, (onewire_trsth*32) need to be larger than (onewire_tpdl_min*4).

onewire_Tdri[7:0] This register is defined the master driving low duration when receiving the rx data from slave, as the diagram in 1.4.2. The duration time will be (1 * onewire_tdri * input_clock_period). For example, if this register is 8'd17 as default and the input clock is 13MHz (76.92ns), the slave tx max bit duration time will be (1*17*0.07692)=1.3 us.

© 2009 MediaTek Inc.

2.19.4 HDQ/1-wire timing requirement

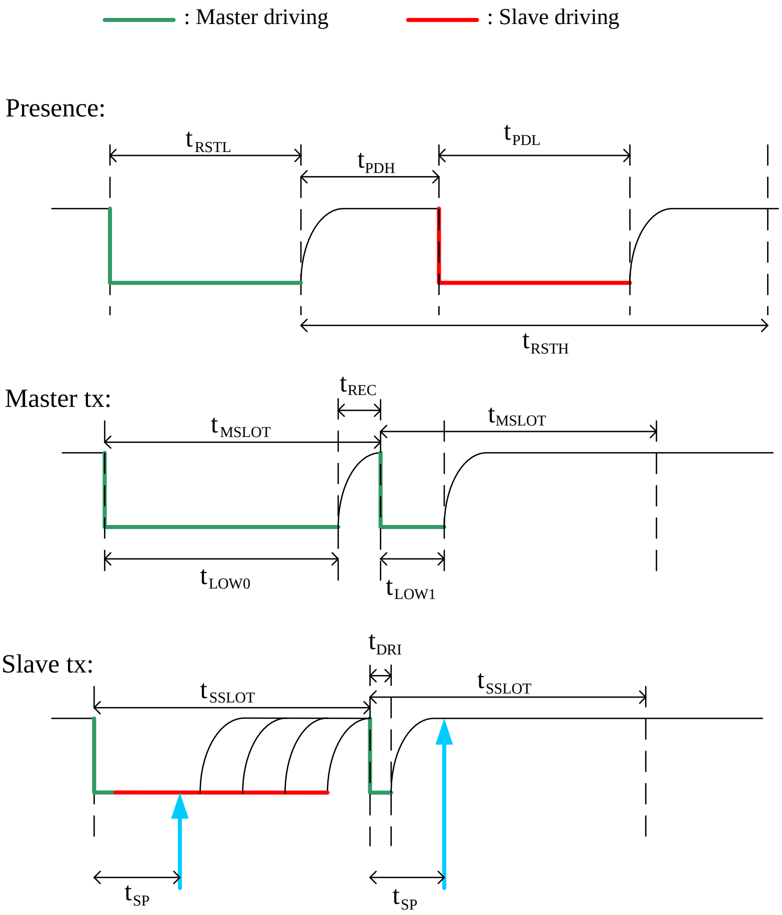
2.19.4.1 HDQ timing requirement



parameter	Max (us)	Type (us)	Mi(us)
t_B		190	
t_{BREAK}		230	
t_{HW1}		40	
t_{HW0}		130	
t_{CYCH}		190	
t_{DW1}	60		10
t_{DW0}	160		70
t_{CYCD}	260		190
t_{SP}		65	

 © 2009 MediaTek Inc.

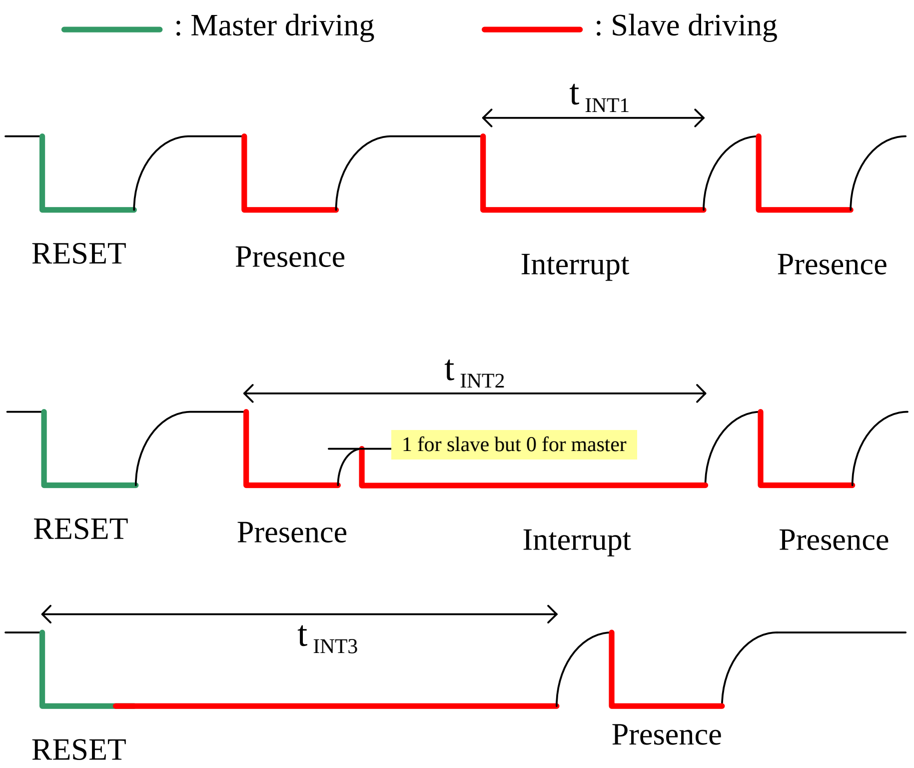
2.19.4.2 1-Wire timing requirement



parameter	Standard Max (us)	Standard Min (us)	Overdrive Max(us)	Overdrive Min(us)
t_{RSTL}		500		50
t_{PDH}	60		6	
t_{PDL}	240	60	24	8
t_{RSTH}		480		48
t_{MSLOT}		100		100
t_{LOW1}		13		1.3
t_{LOW0}		80		8
t_{REC}	15	15	15	15
t_{SSLOT}		120		16
t_{DRI}		1.3		1.3
t_{SP}	110	2	14	2

 © 2009 MediaTek Inc.

2.19.4.3 1-Wire alarm timing



parameter	Standard (us)	Max	Standard (us)	Min	Overdrive Max(us)	Overdrive Min(us)
t_{INT1}	4332		960		433	96
t_{INT2}	4080		1020		408	102
t_{INT3}	4820		960		482	96

2.20 I2C / SCCB Controller

2.20.1 Special Notes

MT6516 has 3 sets of I2C controller: I2C Controller 1 (I2C), I2C Controller 2 (I2C2), I2C Controller 3 (I2C3).

The 3 Controllers are defined as the table below shows:

APB Address Mapping	Pin Name	GPIO MUX
I2C_BASE (8003_0000h)	SCL1 SDA1	SCL1 (GPIO 34) SDA1 (GPIO 35)
I2C2_BASE (8003_5000h)	SCL0 SDA0	none none
I2C3_BASE (8002_E000h)	SCL2 SDA2	SCL2 (GPIO

135) SDA2 (GPIO 136)

Since only Controller 2 has DMA support, all the following descriptions that refers to DMA mode will only be applicable to Controller 2.

2.20.2 General Description

I2C (Inter-IC) /SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

2.20.2.1 Feature Support

I2C compliant master mode operation
Adjustable clock speed for LS/FS mode operation.
7bit/10 bit addressing support.
High Speed mode support.
Slave Clock Extension support.
START/STOP/REPEATED START condition
Manual/DMA Transfer Mode
Multi write per transfer (up to 8 data bytes for non dma mode and 255 data bytes for dma mode)
Multi read per transfer (up to 8 data bytes for non dma mode and 255 data bytes for dma mode)
Multi transfer per transaction (up to 256 write transfers or 256 read transfers with dma mode)
DMA mode with Fifo Flow Control and bus signal holding
Combined format transfer with length change capability.
Active drive / wired-and I/O configuration

2.20.2.2 Manual/DMA Transfer Mode

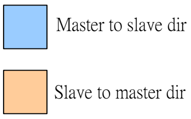
The controller offers 2 types of transfer mode, Manual and DMA.
When Manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows mcu to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.
When DMA mode is enabled, the data to and from the FIFO is controlled via DMA transfer and can therefore support up to 255 bytes of consecutive read or write, with the data read from or write to another memory space. When DMA mode is enabled, flow control mechanism is also implemented to hold the bus clk when FIFO underflow or overflow condition is encountered.

2.20.2.3 Transfer format support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:
(Wording convention note:
transfer = anything encapsulated within a Start and Stop or Repeated Start.

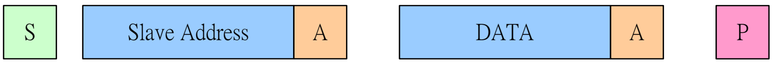
transfer length = the number of bytes within the transfer.
transaction = this is the top unit. Everything combined equals 1 transaction.
Transaction length = the number of transfers to be conducted.

)

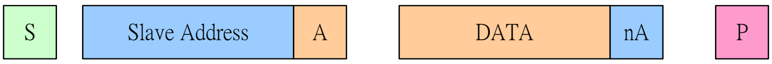


Single Byte Access

Single Byte Write

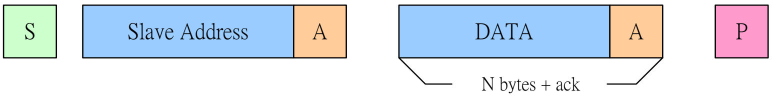


Single Byte Read

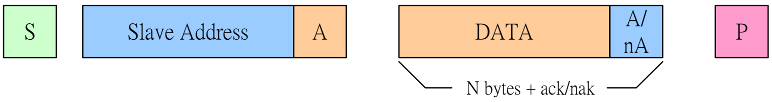


Multi Byte Access

Multi Byte Write

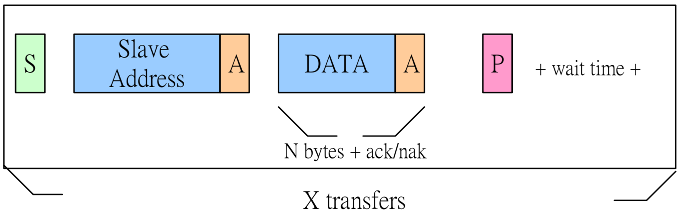


Multi Byte Read

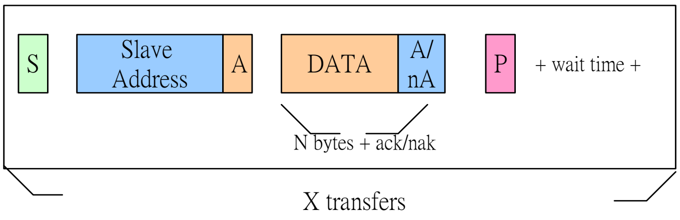


Multi Byte Transfer + Multi Transfer (same direction)

Multi Byte Write + Multi Transfer



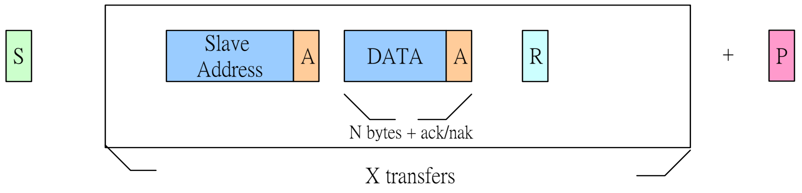
Multi Byte Read + Multi Transfer



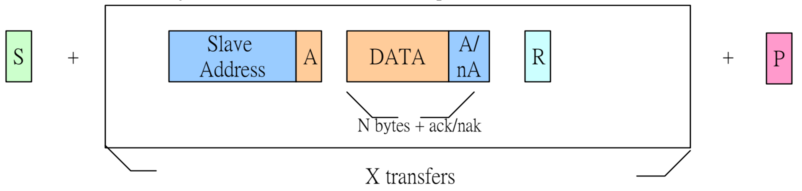
Multi Byte Transfer + Multi Transfer w RS (same direction)

© 2009 MediaTek Inc.

Multi Byte Write + Multi Transfer + Repeated Start



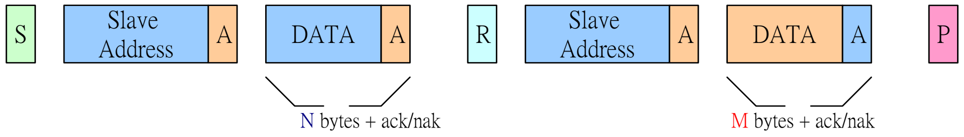
Multi Byte Read + Multi Transfer + Repeated Start



Combined Write/Read with Repeated Start (direction change)

(Note: Only supports Write and then Read sequence. Read and then Write is not supported)

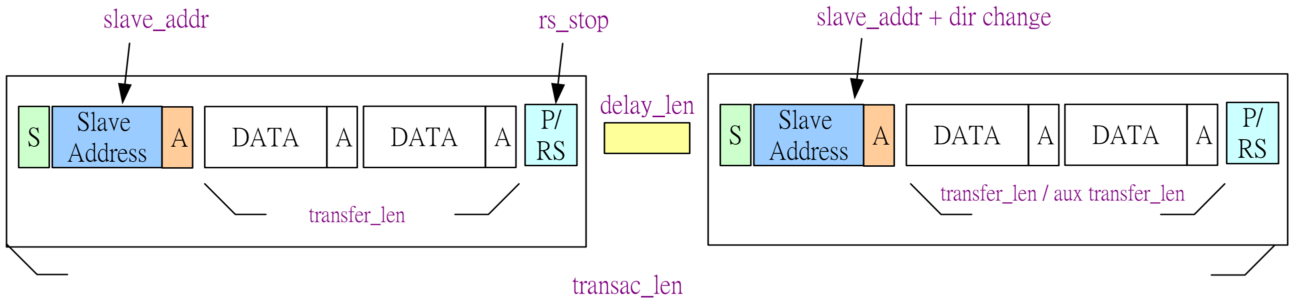
Combined Multi Byte Write + Multi Byte Read



2.20.3 Programming Examples

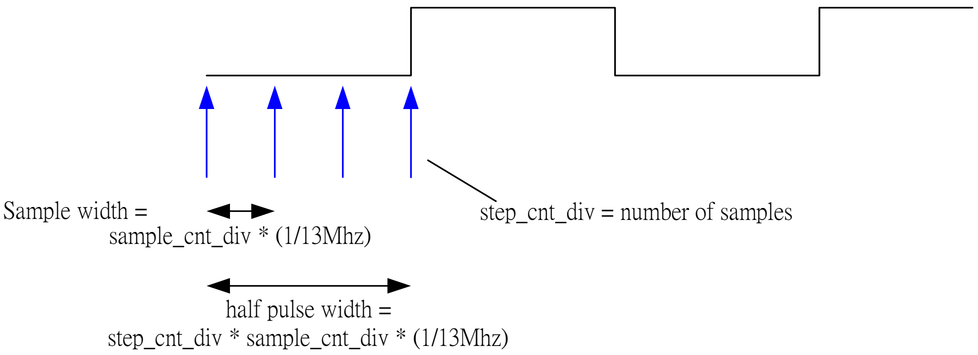
Common Transfer Programmable Parameters

Programmable Parameters



Output Waveform Timing Programmable Parameters

 © 2009 MediaTek Inc.



2.20.4 Register Definitions

I2CREG+0000h Data Port Register DATA_PORT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO DATA**							
Type									R/W							
Reset									0							

DATA_PORT[7:0] This is the FIFO access port. During master write sequences (slave_addr[0] = 0), this port can be written by APB, and during master read sequences (slave_addr[0] = 1), this port can be read by APB.
(NOTE) Slave_addr must be set correctly before accessing the fifo.
(DEBUG ONLY) If the fifo_apb_debug bit is set, then the FIFO can be read and write by the APB

I2CREG+0004h Slave Address Register SLAVE_ADDR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SLAVE_ADDR**							
Type									R/W							
Reset									0							

SLAVE_ADDR [7:0] This specifies the slave address of the device to be accessed. Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer. 1 = master read, 0 = master write.

I2CREG+0008h Interrupt Mask Register INTR_MASK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**DEBUG**	**HS_NACKER**	**ACKERR**	**TRANSAC_COMP**
Type													R.W	R/W	R/W	R/W
Reset													1	1	1	1

This register provides masks for the corresponding interrupt sources as indicated in intr_stat register.

1 = allow interrupt

0 = disable interrupt

Note: while disabled, the corresponding interrupt will not be asserted, however the intr_stat will still be updated with the status. Ie. mask does not affect intr_stat register values.

I2CREG+000Ch Interrupt Status Register INTR_STAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														HS_NACKERR	ACKERR	TRANSAC_COMP
Type														W1C	W1C	W1C
Reset														0	0	0

When an interrupt is issued by i2c controller, this register will need to be read by mcu to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be write 1 cleared.

HS_NACKERR This status is asserted if hs master code nack error detection is enabled. If enabled, hs master code nack err will cause transaction to end and stop will be issued.

ACKERR This status is asserted if ACK error detection is enabled. If enabled, ackerr will cause transaction to end and stop will be issued.

TRANSAC_COMP This status is asserted when a transaction has completed successfully.

I2CREG+0010h Control Register CONTROL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										TRANSFER_LEN_CHANGE	ACKERR_DET_EN	DIR_CHANGE	CLK_EXT EN	DMA_EN	RS_STOP	
Type										R/W	R/W	RW	RW	RW	RW	R/W
Reset										0	0	0	0	0	0	0

TRANSFER_LEN_CHANGE This options specifies whether or not to change the transfer length after the fist transfer completes. If enabled, the transfers after the first transfer will use the transfer_len_aux parameter.

ACKERR_DET_EN This option enables slave ack error detection. When enabled, if slave ack error is detected, the master shall terminate the transaction by issuing a STOP condition and then asserts ackerr interrupt. Mcu shall handle this case appropriately and then resets the fifo address before reissuing transaction again. If this option is disabled, the controller will ignore slave ack error and keep on scheduled transaction.

0 disable

1 enable

DIR_CHANGE This option is used for combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition. Note: when set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.

0 disable

1 enable

CLK_EXT_EN I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing. Therefore, if this bit is set to 1, master controller will enter a high wait state until the slave releases the SCL line.

DMA_EN By default, this is disabled, and fifo data shall be manually prepared by mcu. This default setting should be used for transfer sizes of less than 8 data bytes and no multiple transfer is configured. When enabled, dma requests are turned on, and the fifo data should be prepared in memory.

RS_STOP In LS/FS mode, this bit affects multi-transfer transaction only. It controls whether or not REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP.

In HS mode, this bit must be set to 1.

0 use STOP

1 use REPEATED-START

I2CREG+0014h Transfer Length Register (Number of Bytesper Transfer) TRANSFER_LEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				TRANSFER_LEN_AUX					TRANSFER_LEN							
Type				R/W												R/W
Reset				'h1												'h1

TRANSFER_LEN_AUX[4:0] This field is valid only when dir_change is set to 1. This indicates the number of DATA BYTES to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change. I.e., if dir_change =1, then the first write transfer length depends on transfer_len, while the second read transfer length depend on transfer_len_aux. Dir change is always after the first transfer.

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

TRANSFER_LEN[7:0] This indicates the number of DATA BYTES to be transferred in 1 transfer unit (excluding slave address byte)

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

I2CREG+0018h Transaction Length Register (Number of Transfers per Transaction) TRANSAC_LEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TRANSAC_LEN							
Type																R/W
Reset																'h1

TRANSAC_LEN[7:0] This indicates the number of TRANSFERS to be transferred in 1 transaction

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

I2CREG+001Ch Inter Delay Length Register DELAY_LEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DELAY_LEN							

Type									R/W							
Reset									'h2							

DELAY_LEN[3:0] This sets the wait delay between consecutive transfers when RS_STOP bit is set to 0. (the unit is same as the half pulse width)

I2CREG+0020h Timing Control Register TIMING

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA READ ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	R/W	R/W				R/W					R/W					
Reset	'h0	'h1				'h3					'h3					

LS/FS only. This register is used to control the output waveform timing. Each half pulse width (ie. each high or low pulse) is equal to = step_cnt_div * (sample_cnt_div * 1/13Mhz)

SAMPLE_CNT_DIV[2:0] Used for LS/FS only. This adjusts the width of each sample. (sample width = sample_cnt_div * 1/13Mhz)

STEP_CNT_DIV[5:0] This specifies the number of samples per half pulse width (ie. each high or low pulse)

DATA_READ_ADJ When set to 1, data latch in sampling time during master reads are adjusted according to DATA_READ_TIME value. Otherwise, by default, data is latched in at half of the high pulse width point. This value must be set to less or equal to half the high pulse width.

DATA_READ_TIME[2:0] This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that data is latched in at earlier sampling points (assuming data is settled by then)

I2CREG+0024h Start Register START

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																R/W
Reset																0

START This register starts the transaction on the bus. It is auto deasserted at the end of the transaction.

I2CREG+0030h Fifo Status Register FIFO_STAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_ADDR				WR_ADDR				FIFO_OFFSET						WR_FULL	RD_EMPTY
Type	RO				RO				RO						RO	RO
Reset	0				0				0				0	0	0	0

RD_ADDR[3:0] The current rd address pointer. (only bit [2:0] has physical meaning)

WR_ADDR[3:0] The current wr address pointer. (only bit [2:0] has physical meaning)

FIFO_OFFSET[3:0] wr_addr[3:0] – rd_addr[3:0]

WR_FULL This indicates that the fifo is full.

RD_EMPTY This indicates that the fifo is empty.

I2CREG+0034h Fifo Thresh Register FIFO_THRESH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TX_TRIG_THRESH								RX_TRIG_THRESH		
Type						RW								R/W		
Reset						'h7								'h0		

DEBUG ONLY. By default, these values do not need to be adjusted. Note! for RX, no timeout mechanism is implemented. Therefore, RX_trig_thresh must be left at 0, or there would be data left in the fifo that is not fetched by DMA controller.

TX_TRIG_THRESH[2:0] When tx fifo level is below this value, tx dma request is asserted.

RX_TRIG_THRESH[2:0] When rx fifo level is above this value, rx dma request is asserted.

I2CREG+0038h Fifo Address Clear Register FIFO_ADDR_CLR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FIFO_ADDR_CR
Type																WO
Reset																0

FIFO_ADDR_CLR When written with a 1'b1, a 1 pulse fifo_addr_clr is generated to clear the fifo address to back to 0.

I2CREG+0040h IO Config Register IO_CONFIG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														IO SYNC EN	SDA_IO_CONFIG	SCL_IO_CONFIG
Type														R/W	R/W	R/W
Reset														0	0	0

This register is used to configure the I/O for the sda and scl lines to select between normal i/o mode, or open-drain mode to support wired-and bus.

IO_SYNC_EN DEBUG ONLY: When set to 1, scl and sda inputs will be first dual synced by bclk_ck. This should not be needed. Only reserved for debugging.

SDA_IO_CONFIG
0 normal tristate io mode
1 open-drain mode

SCL_IO_CONFIG
0 normal tristate io mode
1 open-drain mode

I2CREG+0044h RESERVED DEBUG Register DEBUG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type														R/W	R/W	R/W
Reset														0	0	0

NOTE: This register is for DEBUG ONLY. The bits are R/W, do not change the values from the default value.

I2CREG+0048h High Speed Mode Register HS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SAMPLE_CNT DIV				HS_STEP_CNT_DIV				MASTER_CODE					HS_NACKERR_DET_EN	HS_EN
Type		R/W				R/W				R/W					R/W	R/W
Reset		0				1				0					1	0

This register contains options for supporting high speed operation features

Each HS half pulse width (ie. each high or low pulse) is equal to = step_cnt_div * (sample_cnt_div * 1/13Mhz)

HS_SAMPLE_CNT_DIV[2:0] When high speed mode is entered after the master code transfer has been completed, the sample width becomes dependent on this parameter.

HS_STEP_CNT_DIV[2:0] When high speed mode is entered after the master code transfer has been completed, the number of samples per half pulse width becomes dependent on this value.

MASTER_CODE[2:0] This is the 3 bit programmable value for the master code to be transmitted.

HS_NACKERR_DET_EN This enables NACKERR detection during the master code transmission. When enabled, if NACK is not received after master code has been transmitted, the transaction will terminated with a STOP condition.

HS_EN This enables the high speed transaction. (note: rs_stop must be set to 1 as well)

I2CREG+0050h Soft Reset Register SOFTRESET

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

SOFT_RESET When written with a 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

I2CREG+0064h Debug Status Register DEBUGSTAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE			
Type										RO	RO	RO	RO			
Reset										0	1	0	0			

BUS_BUSY DEBUG ONLY: valid when bus_detect_en is 1. bus_busy = 1 indicates a start transaction has been detected and no stop condition has been detected yet.

MASTER_WRITE DEBUG ONLY: 1 = current transfer is in the master write dir

MASTER_READ DEBUG ONLY: 1 = current transfer is in the master read dir

MASTER_STATE[3:0] DEBUG ONLY: reads back the current master_state.

© 2009 MediaTek Inc.

I2CREG+0068h Debug Control Register **DEBUGCTRL**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBUG_RD	FIFO_APB_DEBUG
Type															WO	R/W
Reset															0	0

APB_DEBUG_RD This bit is only valid when fifo_apb_debug is set to 1. Writing to this register will generate a 1 pulsed fifo apb rd signal for reading the fifo data.

FIFO_APB_DEBUG This is used for trace32 debug purposes. When using trace32, and the memory map is shown, turning this bit on will block the normal apb read access. Apb read access to the fifo is then enabled by writing to apb_debug_rd.

0 disable

1 enable

2.21 IrDA Framer

2.21.1 General Description

IrDA framer is implemented to reduce the CPU loading for IrDA transmissions by performing all the physical level protocol framing in hardware. From a software perspective, the framer need only prepare and process the raw data for transmission and reception. Generic DMA is required to move the data between IrDA framer's internal FIFO and software-designated memory. The IrDA framer supports IrDA SIR, MIR, and FIR modes of operation. SIR mode includes operation from 9600bps ~ 115200bps, MIR includes operation at 567000bps or 1152000bps, and FIR mode includes operation at 4Mbps.

2.21.2 Register Definitions

IRDA+0000h TX BUF and RX BUF BUF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUF[7:0]							
Type									R/W							
Reset									0							

BUF IrDA Framer transmit or receive data.

A write to this register writes into the internal TX FIFO.

A read from this register reads from the internal RX FIFO.

IRDA+0004h TX BUF and RX BUF clear signal BUF_CLEAR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CLEAR
Type																R/W
Reset																0

CLEAR **SIR mode only**. When CLEAR=1, both the TX and RX FIFO are cleared. This is used primarily for debug purpose. Normal operation does not require this. This control signaled can only be issued under SIR mode.

IRDA+0008h Maximum Turn Around Time MAX_T

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**MAX_T [13:0]**													
Type			R/W													
Reset			3E80h													

MAX_T The maximum time that a station can hold the P/F bit. This parameter along with the baud rate parameter dictates the maximum number of bytes that a station can transmit before passing the line to another station by transmitting a frame with the P/F bit. This parameter is used by one station to indicate the maximum time the other station can send before it must turn the link around. For baud rates less than 115200 kbps, 500 ms is the only valid value. The default value is 500 ms.

IRDA+000Ch Minimum Turn Around Time MIN_T

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MIN_T [15:0]**															
Type	R/W															
Reset	FDE8h															

MIN_T Minimum turn around time, the default value is 10 ms. The minimum turn around time parameter deals with the time needed for a receiver to recover following saturation by transmission from the same device. This parameter corresponds to the required time delay between the last byte of the last frame sent by a station and the point at which it is ready to receive the first byte of a frame from another station, i.e. the latency for a transmit to complete and be ready to receive.

IRDA+0010h Number of additional BOFs prefixed to the beginning of a frame BOFS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**TYPE**	**BOFS [6:0]**						
Type									R/W	R/W						
Reset									0	1011b						

BOFs **For SIR mode**: the additional BOFs parameter indicates the number of additional flags needed at the beginning of every frame. The main purpose for the additional BOFs is to provide a delay at the beginning of each frame for devices with a long interrupt latency.

For MIR mode: This parameter indicates the number of double STA's to transmit in the beginning. This value should be set to 0 (for default 2 STA's) for MIR mode, unless more are required.

For FIR mode: This parameter has no effect.

TYPE **SIR mode only**. Additional BOFs type.

1 BOF = C0h

0 BOF = FFh

IRDA+0014h Baud rate divisor DIV

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIV[15:0]**															
Type	R/W															

Reset	55h

DIV Transmit or receive rate divider. Rate = System clock frequency / DIV/ 16. The default value is 55h when in contention mode. **This divisor is also used to determine the RX FIFO timeout threshold.**

IRDA+0018h Transmit frame size TX_FRAME_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**TX_FRAME_SIZE[11:0]**											
Type					R/W											
Reset					40h											

TX_FRAME_SIZE Transmit frame size; the default value is 64 when in contention mode.

IRDA+001Ch Receiving frame1 size RX_FRAME1_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**RX_FRAME1_SIZE[11:0]**											
Type					RO											
Reset					0											

RX_FRAME1_SIZE Reports the number of byte received. Includes only the A+C+I fields.

IRDA+0020h Transmit abort indication ABORT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ABORT**
Type																R/W
Reset																0

ABORT **SIR mode only**. When set 1, the framer transmits an abort sequence and closes the frame without an FCS field or an ending flag.
Note: Tx abort can be achieved in MIR and FIR by simply disabling the tx_en signal.

IRDA+0024h IrDA framer transmit enable signal TX_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**TX_ONE**	**TXINVERT**	**MODE**	**TX_EN**
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

TX_EN Transmit enable.

MODE **SIR mode only**. Modulation type selection.

0 3/16 modulation

1 1.61us

TXINVERT Invert the transmit signal.

0 Transmit signal is not inverted.

1 Transmit signal is inverted.

TX_ONE: Controls the transmit enable signal is one or not.

0 tx_en is not de-asserted until software programs a so.

1 tx_en is de-asserted (i.e. transmit disabled) automatically after one frame has been sent.

IRDA+0028h IrDA framer receive enable signal RX_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**RX_ONE**	**RXINVERT**	**RX_EN**
Type														R/W	R/W	R/W
Reset														0	0	0

RX_EN Receive enable.

RXINVERT Invert the receive signal.

- 0 Receive signal is not inverted.
- 1 Receive signal is inverted.

RX_ONE Disable receive when get one frame.

- 0 rx_en is not de-asserted until software programs so.
- 1 rx_en is de-asserted (i.e. transmit disabled) automatically after one frame has been sent.

IRDA+002Ch FIFO trigger level indication TRIGGER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**RX_TRIG[**		**TX_TRIG**	
Type													R/W		R/W	
Reset													0		0	

TX_TRIG TX FIFO interrupt trigger threshold. When the amount of data in the TX FIFO is less than the specified amount, dma req is asserted. (When TX_TRIG = 03, dma req is always asserted as long as FIFO is not full.)

- 00 0 byte
- 01 1 byte
- 02 8 byte
- 03 16 byte

RX_TRIG RX FIFO interrupt trigger threshold. When the amount of data in RX FIFO is above the specified amount, dma req is asserted.

- 00 1 byte
- 01 2 byte
- 02 3 byte

IRDA+0030h IRQ enable signal IRQ_ENABLE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				2NDRX_COMP	RXRESTART	THRESHTIMEOUT	FIFOTIMEOUT	TXABORT	RXABORT	MAXTIMEOUT	MINTIMEOUT	RXCOMPLETE	TXCOMPLETE	ERROR	RXTHRES	TXTHRES
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0

TXRES Transmit data reaches the threshold level. (For debug only. Should be set to 0.)

- 0 No interrupt is generated.
- 1 Interrupt is generated when transmit FIFO size reaches threshold.

RXRES Receive data reaches the threshold level. (For debug only. Should be set to 0.)

0 No interrupt is generated.
1 Interrupt is generated when receive FIFO size reaches threshold.

ERROR Error status interrupt enable.
0 No interrupt is generated.
1 Interrupt is generated when one of the error statuses occurs.

TXCOMPLETE Transmit one frame completely.
0 No interrupt is generated.
1 Interrupt is generated when transmitting one frame completely.

RXCOMPLETE Receive one frame completely.
0 No interrupt is generated.
1 Interrupt is generated when receiving one frame completely.

MINTIMEOUT Minimum time timeout.
0 No interrupt is generated.
1 Interrupt is generated when minimum timer is timed out.

MAXTIMEOUT Maximum time timeout.
0 No interrupt is generated.
1 Interrupt is generated when maximum timer is timed out.

RXABORT Receiving aborting frame.
0 No interrupt is generated.
1 Interrupt is generated when receiving aborting frame.

TXABORT **SIR mode only**. Transmitting aborting frame.
0 No interrupt is generated.
1 Interrupt is generated when transmitting aborting frame.

FIFOTIMEOUT FIFO timeout.
0 No interrupt is generated.
1 Interrupt is generated when FIFO timeout.

THRESHTIMEOUT Threshold time timeout.
0 No interrupt is generated.
1 Interrupt is generated when threshold timer is timed out.

RXRESTART **SIR mode only**. Receiving a new frame before one frame is received completely.
0 No interrupt is generated.
1 Interrupt is generated when receiving a new frame before one frame is received completely.

2NDRX_COMP Receiving second frame and get P/F bit.
0 No interrupt is generated.
1 Interrupt is generated when receiving second frame and get P/F bit completely.

IRDA+0034h Interrupt Status IRQ_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				2NDRX_COMP	RXRESTART	THRESHTIMEOUT	FIFOTIMEOUT	TXABORT	RXABORT	MAXTIMEOUT	MINTIMEOUT	RXCOMPLETE	TXCOMPLETE	ERROR	RXTRES	TXTRES
Type				RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

TXFIFO Transmit FIFO reaches threshold. (For debug only. Not recommended for normal usage.)
RXFIFO Receive FIFO reaches threshold. (For debug only. Not recommended for normal usage.)
ERROR Generated when any of status in Error Status register occurs.

Once the source of an interrupt is determined to be caused by an error (bit 2), the error status register should be read. Once read, both the error status register and the interrupt source are read-cleared. If the error status register indicates either a frame 1 or frame 2 error, the corresponding frame status register should be read.

TXCOMPLETE Transmitting one frame completely.
RXCOMPLETE Receiving one frame completely.
MINTIMEOUT Minimum turn around time timeout.
MAXTIMEOUT Maximum turn around time timeout.
RXABORT Receiving aborting frame.
TXABORT Transmitting aborting frame.
FIFOTIMEOUT FIFO is timeout.
THRESHTIMEOUT Threshold time timeout.
RXRESTART Receiving a new frame before one frame is received completely.
2NDRX_COMP Receiving second frame and get P/F bit completely.

IRDA+0038h ERROR STATUS register ERR_STATUS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**TX FIFO UNDERRUN**	**FRAME 2 DATA ERR**	**FRAME 1 DATA ERR**	RESERVED2	RESERVED	OVERRUN	RXSIZE
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

RXSIZE Receive frame size error.
OVERRUN Frame overrun.
RESERVED Reserved for future use.
RESERVED2 Reserved for future use.
FRAME1 DATA ERR Indicates that an error condition occurred in RX frame1. Must check the RX frame1 status.
FRAME2 DATA ERR Indicates that an error condition occurred in RX frame2. Must check the RX frame2 status.
TX FIFO UNDERRUN **MIR and FIR mode only**.

TX FIFO underrun has occurred. Data transmission is aborted. Software must reset the tx_en signal.

IRDA+003Ch Transceiver power on/off control. Transceiver mode select. TRANSCEIVER_PDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TXCVR CONFIG	TX MANUAL	TRANS_PDN
Type														R/W	R/W	R/W
Reset														0	0	0

TRANSCEIVER_PDN Used for power on/off control for external IrDA transceiver.

TX_MANUAL When txcvr config is set to 1, this bit can be used to select the operation mode of the external IrDA transceiver (some transceivers require selection between high speed and low speed operating modes), by software programming the desired sequence to transmit through the irda_txd pin.

TXCVR CONFIG

- 0 Irda_txd comes from core logic.
- 1 Irda_txd depends on tx_manual value.

IRDA+0040h Maximum number of receiving frame size RX_FRAME_MAX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**MAX_RX_FRAME_SIZE_**											
Type					R/W											
Reset					0											

RX_FRAME_MAX Receive frame I field max size, when actual receiving frame size is larger than rx_frame_max, RXSIZE is asserted. The maximum allowed I field size is 2048.

IRDA+0044h Threshold Time THRESH_T

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DISCONNECT_TIME[15:0]**															
Type	R/W															
Reset	bb8h															

THRESHOLD TIME Threshold time; used to control the time a station waits without receiving a valid frame before disconnecting the link. Associated with this is the time a station waits without receiving a valid frame before sending a status indication to the service user layer.

IRDA+0048h Counter enable signal COUNT_ENABLE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														THRESH_EN	MIN_EN	MAX_EN
Type														R/W	R/W	R/W
Reset														0	0	0

COUNT_ENABLE Counter enable signals.

IRDA+004Ch Indication of system clock rate CLOCK_RATE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**CLOCK_RATE**	
Type															R/W	
Reset															0	

CLOCK_RATE **SIR mode only** Indication of the system clock rate.

- 0 26 MHz
- 1 52 MHz
- 2 13 MHz

IRDA+0050h System Clock Rate Fix RATE_FIX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name												MIR TIMING TUNE		CRC REPORT	SIR FRAMING SET	RATE_FIX
Type												R/W		R/W	R/W	R/W
Reset												0		0	0	0

RATE_FIX **SIR mode only** Fix the IrDA framer sample base clock rate as 13 MHz.

0 Clock rate based on clock_rate selection.

1 Clock rate fixed at 13 MHz.

SIR FRAMING SET **SIR mode only**. Framing error check condition.

0 Ignore the STOP bit of the last byte of a frame.

1 Check the STOP bit of the last byte of a frame.

CRC REPORT When set to 1, CRC error is reported via error status register and error interrupt.

MIR TIMING TUNE[1:0] **MIR mode only**. For some transceivers, in MIR 0.576mbps mode, the RX output pulse does not conform to IRDA specification. Therefore, this option is used to detect the RX output from those transceivers correctly.

0 For transceivers that conform to spec.

1 For transceivers that do not conform to spec, and the RX output pulse is half of that specified.

2 For transceivers that do not conform to spec, and the RX output pulse is quarter of that specified.

IRDA+0054h RX Frame1 Status FRAME1_STATUS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FIR STO ERR	FIR 4PPM ERR	MIR HDLC ERR	UNKNOW_ERROR	PF_DETECT	CRC_FAIL	FRAME_ERROR
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

FRAME_ERROR **SIR mode only**. Framing error, i.e. STOP bit = 0.

0 No framing error

1 Framing error occurred

CRC_FAIL CRC check fail

2 CRC check successfully

3 CRC check fail

PF_DETECT P/F bit detect

0 Not a P/F bit frame

1 Detected P/F bit in this frame

UNKNOWN_ERROR **SIR mode only**. Receiving error data, i.e. escape character is followed by a character that is not an ESC, BOF, or EOF character.

0 Data received correctly.

1 Unknown error occurred.

MIR HDLC ERR **MIR mode only**. MIR HDLC encoding error
- 0 No error
- 1 Error

FIR 4PPM ERR **FIR mode only**. FIR 4ppm encoding error
- 0 No error
- 1 Error

FIR STO ERR **FIR mode only**. FIR STO sequence error
- 0 No error
- 1 Error

IRDA+0058h RX Frame2 Status FRAME2_STATUS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FIR STO ERR	FIR 4PPM ERR	MIR HDLC ERR	UNKNOW_ERROR	PF_DETECT	CRC_FAIL	FRAME_ERROR
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

FRAME_ERROR **SIR mode only.** Framing error, i.e. STOP bit = 0
- 0 No framing error.
- 1 Framing error occurred.

CRC_FAIL CRC check fail.
- 0 CRC check successfully.
- 1 CRC check fail.

PF_DETECT P/F bit detect.
- 0 Not a P/F bit frame.
- 1 Detected P/F bit in this frame.

UNKNOWN_ERROR **SIR mode only**. Receiving error data, i.e. escape character is followed by a character that is not an ESC, BOF, or EOF character.
- 0 Data receiving correctly.
- 1 Unknown error occurred.

MIR HDLC ERR **MIR mode only**. MIR HDLC encoding error.
- 0 No error
- 1 Error

FIR 4PPM ERR **FIR mode only**.FIR 4ppm encoding error
- 0 No error
- 1 Error

FIR STO ERR **FIR mode only**.FIR STO sequence error
- 0 No error
- 1 Error

IRDA+005Ch Receiving frame2 size RX_FRAME2_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					RX_FRAME2_SIZE[11:0]											
Type					RO											

Reset					0											

RX_FRAME2_SIZE Reports the number of byte received. Includes only the A+C+I fields.

IRDA+0060h Irda Mode Select IRDA_MODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														MIR SPEED	IRDA MODE	
Type														R/W	R/W	
Reset														0	00	

IRDA MODE Selects the IrDA operating mode. NOTE: this mode selection cannot be issued while transmitting or receiving.

- 00 IR mode
- **01** MIR mode
- **10** FIR mode

MIR SPEED Select the MIR speed.

- 0 0.576 Mbps
- 1 1.152 Mbps

IRDA+0064h Fifo Status FIFO_STAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												RX FIFO HOLD	TX FIFO WR FULL	TX FIFO RD EMPTY	RX FIFO WR FULL	RX FIFO RD EMPTY
Type												RO	RO	RO	RO	RO
Reset												0	0	1	0	1

This register indicates the real time FIFO status, for monitoring purposes.

2.22 Keypad Scanner

2.22.1 General Description

The keypad can be divided into two parts: one is the keypad interface including 8 columns and 8 rows with one dedicated power-key, as shown in **Fig. 3**; the other is the key detection block which provides key pressed, key released and de-bounce mechanisms. Each time the key is pressed or released, i.e. something different in the 8 x 8 matrix or power-key, the key detection block senses the change and recognizes if a key has been pressed or released. Whenever the key status changes and is stable, a KEYPAD IRQ is issued. The MCU can then read the key(s) pressed directly in KP_MEM1, KP_MEM2, KP_MEM3, KP_MEM4 and KP_MEM5 registers. To ensure that the key pressed information is not missed, the status register in keypad is not read-cleared by APB read command. The status register can only be changed by the key-pressed detection FSM.

This keypad can detect one or two key-pressed simultaneously with any combination. **Fig. 4** shows one key pressed condition. **Fig. 5** (a) and **Fig. 5** (b) illustrate two keys pressed cases. Since the key press detection depends on the HIGH or LOW level of the external keypad interface, if keys are pressed at the same time and there exists a key that is on the same column and the same row with the other keys, the pressed key cannot be correctly decoded. For example, if there are three key presses: key1 = (x1, y1), key2 = (x2, y2), and key3 = (x1, y2), then both key3 and key4 = (x2, y1) are detected, and therefore they cannot be distinguished correctly. Hence, the keypad can detect only one or two keys pressed simultaneously at any combination. More than two keys pressed simultaneously in a specific pattern retrieve the wrong information.

This keypad can detect more than two key-presses simultaneously with some special combination. This limitation is any two or more simultaneous keys can't be the same row or column. As shown in **Fig.4**, two key-presses has exists in the keypad matrix and we do the 3rd key-press, KEY_A or KYE_B. In this case, KEY_A is valid and can be detected correctly because its position is not in the row or column of the two existing key-press. However keypad circuit can't detect KEY_B correctly. If three or more simultaneous key-press is necessary, the positions of these special keys need to be arranged carefully.

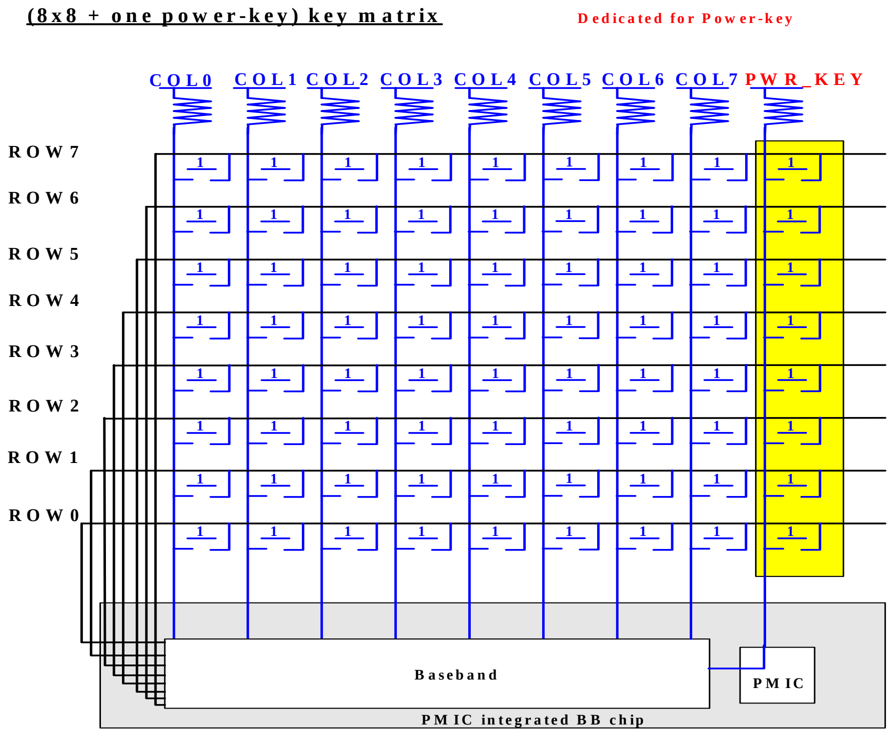


Fig. 3 8x8 matrix with one power-key

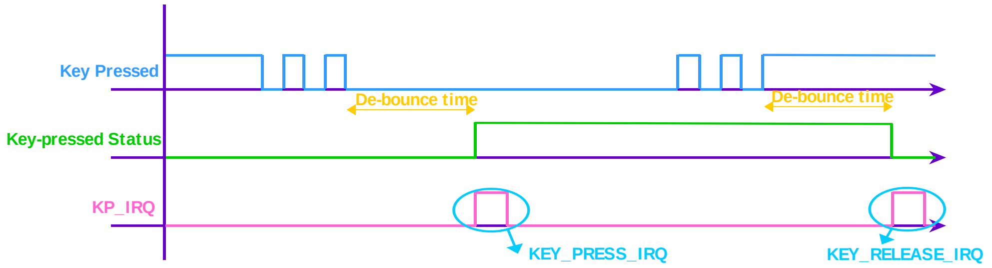


Fig. 4. One key pressed with de-bounce mechanism denoted

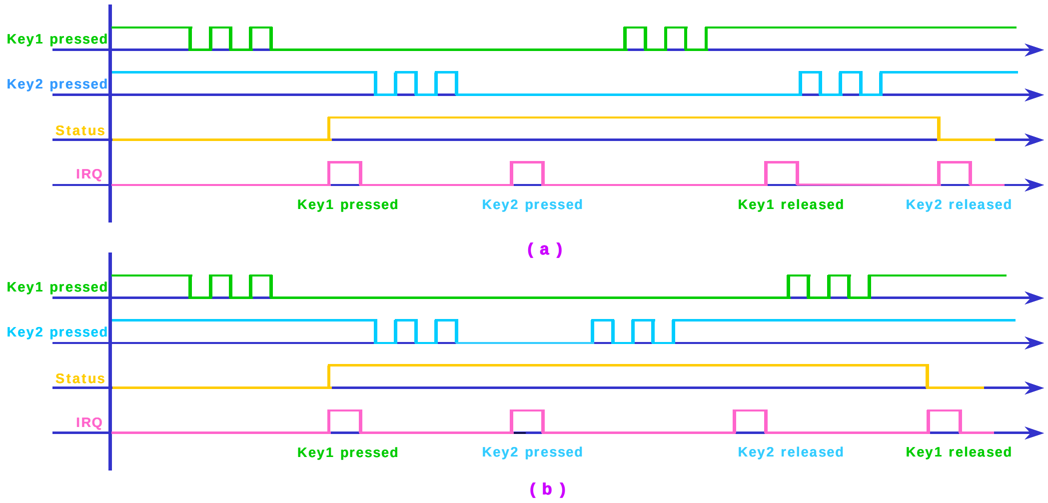


Fig. 5. (a) Two keys pressed, case 1 (b) Two keys pressed, case 2

© 2009 MediaTek Inc.

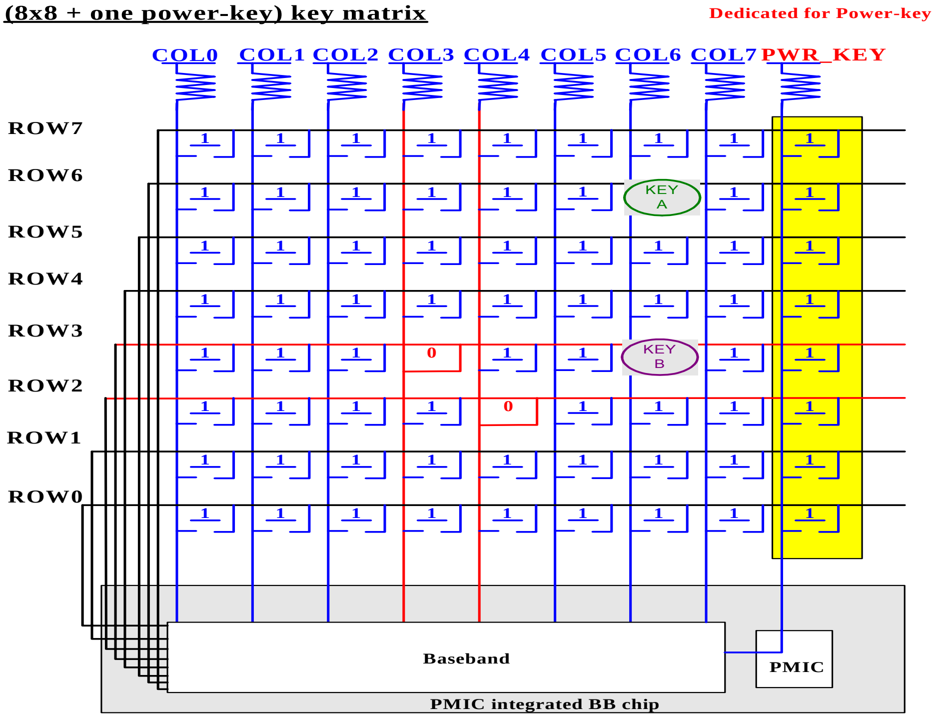


Fig. 6. Simultaneous three key support example.

2.22.2 Register Definitions

KP +0000h Keypad status KP_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STA**
Type																RO
Reset																0

STA This register indicates the keypad status. The register is not cleared by the read operation.

- **0** No key pressed
- **1** Key pressed

KP +0004h Keypad scanning output Register KP_MEM1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY 15**	**KEY 14**	**KEY 13**	**KEY 12**	**KEY 11**	**KEY 10**	**KEY 9**	**KEY 8**	**KEY 7**	**KEY 6**	**KEY 5**	**KEY 4**	**KEY 3**	**KEY 2**	**KEY 1**	**KEY 0**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The register shows up the key-press status of key0(LSB)~key15. Please reference Table 37.

KP +0008h Keypad scanning output Register KP_MEM2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**KEY 31**	**KEY 30**	**KEY 29**	**KEY 28**	**KEY 27**	**KEY 26**	**KEY 25**	**KEY 24**	**KEY 23**	**KEY 22**	**KEY 21**	**KEY 20**	**KEY 19**	**KEY 18**	**KEY 17**	**KEY 16**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The register shows up the key-press status of key16(LSB)~key31. Please reference Table 37.

KP +000Ch Keypad scanning output Register KP_MEM3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY 47**	**KEY 46**	**KEY 45**	**KEY 44**	**KEY 43**	**KEY 42**	**KEY 41**	**KEY 40**	**KEY 39**	**KEY 38**	**KEY 37**	**KEY 36**	**KEY 35**	**KEY 34**	**KEY 33**	**KEY 32**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The register shows up the key-press status of key32(LSB)~key47. Please reference Table 37.

KP +0010h Keypad scanning output Register KP_MEM4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY 63**	**KEY 62**	**KEY 61**	**KEY 60**	**KEY 59**	**KEY 58**	**KEY 57**	**KEY 56**	**KEY 55**	**KEY 54**	**KEY 53**	**KEY 52**	**KEY 51**	**KEY 50**	**KEY 49**	**KEY 48**
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The register shows up the key-press status of key48(LSB)~key63. Please reference Table 37.

KP +0014h Keypad scanning output Register KP_MEM5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**KEY 71**	**KEY 70**	**KEY 69**	**KEY 68**	**KEY 67**	**KEY 66**	**KEY 65**	**KEY 64**
Type									RO	RO	RO	RO	RO	RO	RO	RO
Reset									1	1	1	1	1	1	1	1

The register shows up the key-press status of key64(LSB)~key71. Please reference Table 37.

These five registers list the status of 72 keys on the keypad but KEY[8], KEY[17], KEY[26], KEY[35], KEY[44] , KEY[53] , KEY[62] , KEY[71] is dedicated for power key. When the MCU receives the KEYPAD IRQ, both two registers must be read. If any key is pressed, the relative bit is set to 0.

If some keys can be use because their COL or ROW is use as GPIO, these corresponding bit will be tie to high.

KEYS Status list of the 72 keys.

KP +00018h De-bounce period setting KP_DEBOUNCE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DEBOUNCE [13:0]**													
Type			R/W													
Reset			400h													

This register defines the waiting period before key press or release events are considered stale. If debounce setting is too small, keypad will be too sensitive and detect too many unexpected key-press. The suitable debounce time setting must be adjust for the user's habit.

DEBOUNCE De-bounce time = KP_DEBOUNCE*(1/32k) sec. 32 kHz is the working clock frequency of keypad module.

	COL0	COL1	COL2	COL3	COL4	COL5	COL6	COL7	PWRKEY
ROW7	63	64	65	66	67	68	69	70	71
ROW6	54	55	56	57	58	59	60	61	62
ROW5	45	46	47	48	49	50	51	52	53
ROW4	36	37	38	39	40	41	42	43	44
ROW3	27	28	29	30	31	32	33	34	35
ROW2	18	19	20	21	22	23	24	25	26
ROW1	9	10	11	12	13	14	15	16	17
ROW0	0	1	2	3	4	5	6	7	8

Table 37 KEY's order number in COL/ROW matrix.

Revision	Date	Author	Comments
	2008/7/18	SJ Yang	Add clock rate switching notes

2.23 Memory Stick and SD Memory Card Controller

2.23.1 Introduction

The controller fully supports the Memory Stick bus protocol as defined in Format Specification version 2.0 of Memory Stick Standard (Memory Stick PRO) and the SD Memory Card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0 as well as the MultiMediaCard (MMC) bus protocol as defined in MMC system specification version 4.1. Since SD Memory Card bus protocol is backward compatible to MMC bus protocol, the controller is capable of working well as the host on MMC bus under control of proper firmware. Furthermore, the controller also support SDIO card specification version 1.0 partially. However, the controller can only be configured as either the host of Memory Stick or the host of SD/MMC Memory Card at one time. Hereafter, the controller is also abbreviated as MS/SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Shared pins for Memory Stick and SD/MMC Memory Card
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported

 © 2009 MediaTek Inc.

- Interrupt capabilities
- Automatic command execution capability when an interrupt from Memory Stick
- Data rate up to 26 Mbps in serial mode, 26x4 Mbps (26x8 Mbps if 8-bit data line for SD/MMC card is configured) in parallel model, the module is targeted at 26 MHz operating clock
- Serial clock rate on MS/SD/MMC bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Not support SPI mode for MS/SD/MMC Memory Card
- Not support multiple SD Memory Cards

2.23.2 Overview

2.23.2.1 Pin Assignment

Since the controller can only be configured as either the host of Memory Stick or the host of SD/MMC Memory Card at one time, pins for Memory Stick and SD/MMC Memory Card are shared in order to save pin counts. The following lists pins required for Memory Stick and SD/MMC Memory Card. **Table 38** shows how they are shared. In **Table 38**, all I/O pads have embedded both pull up and pull down resistor because they are shared by both the Memory Stick and SD/MMC Memory Card. Pins 2,4,5,8 are only useful for SD/MMC Memory Card. Pull down resistor for these pins can be used for power saving. All embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers if optimal pull-up or pull-down resistors are required on the system board. The pin VDDPD is used for power saving. Power for Memory Stick or SD/MMC Memory Card can be shut down by programming the corresponding control register. The pin WP (Write Protection) is only valid when the controller is configured for SD/MMC Memory Card. It is used to detect the status of Write Protection Switch on SD/MMC Memory Card.

No.	Name	Type	MMC	SD	MS	MSPRO	Description
1	SD_CLK	O	CLK	CLK	SCLK	SCLK	Clock
2	SD_DAT3	I/O/PP		CD/DAT3		DAT3	Data Line [Bit 3]
3	SD_DAT0	I/O/PP	DAT0	DAT0	SDIO	DAT0	Data Line [Bit 0]
4	SD_DAT1	I/O/PP		DAT1		DAT1	Data Line [Bit 1]
5	SD_DAT2	I/O/PP		DAT2		DAT2	Data Line [Bit 2]
6	SD_CMD	I/O/PP	CMD	CMD	BS	BS	Command Or Bus State
7	SD_PWRON	O					VDD ON/OFF
8	SD_WP	I					Write Protection Switch in SD
9	SD_INS	I	VSS2	VSS2	INS	INS	Card Detection

Table 38 ***Sharing of pins for Memory Stick and SD/MMC Memory Card Controller***

2.23.2.2 Card Detection

For Memory Stick, the host or connector should provide a pull up resistor on the signal INS. Therefore, the signal INS will be logic high if no Memory Stick is on line. The scenario of card detection for Memory Stick is shown in **Figure 19**. Before Memory Stick is inserted or powered on, on host side SW1 shall be closed and SW2 shall be opened for card detection. It is the default setting when the controller is powered on. Upon insertion of Memory Stick, the signal INS will have a transition from high to low. Hereafter, if Memory Stick is

removed then the signal INS will return to logic high. If card insertion is intended to not be supported, SW1 shall be opened and SW2 closed always.

For SD/MMC Memory Card, detection of card insertion/removal by hardware is also supported. Because a pull down resistor with about 470 KΩ resistance which is impractical to embed in an I/O pad is needed on the signal CD/DAT3, and it has to be capable of being connected or disconnected dynamically onto the signal CD during initialization period, an additional I/O pad is needed to switch on/off the pull down resistor on the system board. The scenario of card detection for SD/MMC Memory Card is shown in **Figure 20**. Before SD/MMC Memory Card is inserted or powered on, SW1 and SW2 shall be opened for card detection on the host side. Meanwhile, pull down resistor R_{CD} on system board shall attach onto the signal CD/DAT3 by the output signal RCDEN. In addition, SW3 on the card is default to be closed. Upon insertion of SD/MMC Memory Card, the signal CD/DAT3 will have a transition from low to high. If SD/MMC Memory Card is removed then the signal CD/DAT3 will return to logic low. After the card identification process, pull down resistor R_{CD} on system board shall disconnect with the signal CD/DAT3 and SW3 on the card shall be opened for normal operation.

Since the scheme above needs a mechanical switch such as a relay on system board, it is not ideal enough. Thus, a dedicated pin “INS” is used to perform card insertion and removal for SD/MMC. The pin “INS” will connect to the pin “VSS2” of a SD/MMC connector. Then the scheme of card detection is the same as that for MS. It is shown in **Figure 19**.

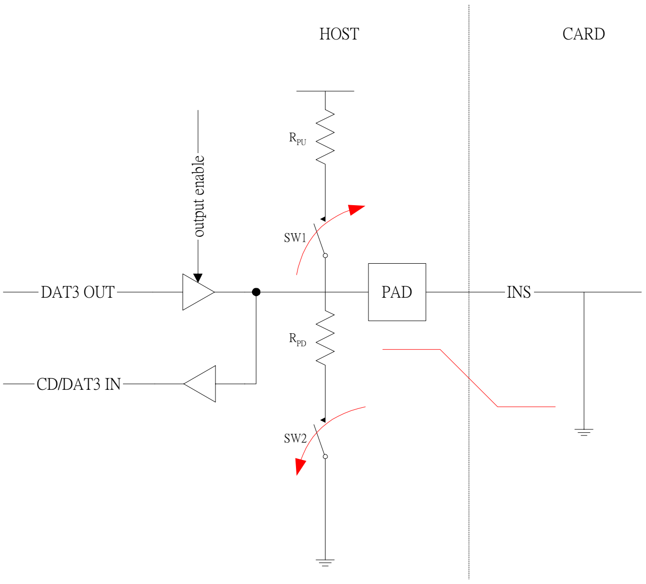


Figure 19 ***Card detection for Memory Stick***

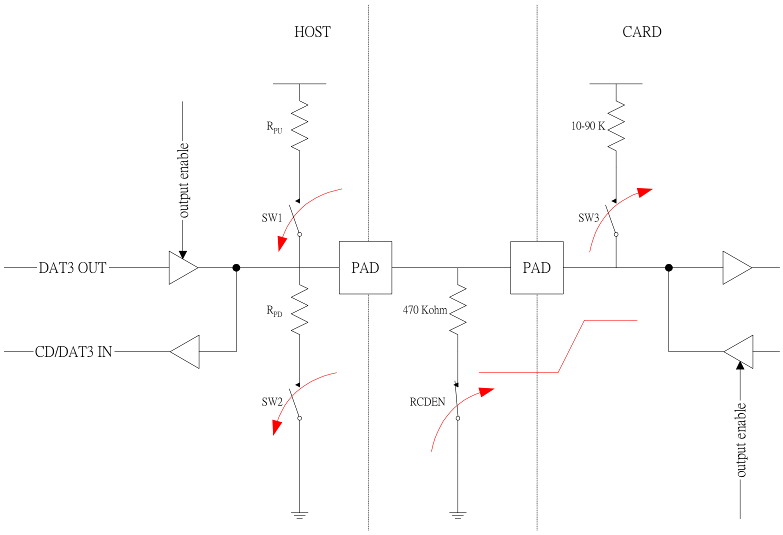


Figure 20 ***Card detection for SD/MMC Memory Card***

2.23.3 Register Definitions

For MT6253, MSDC base address is 0x81110000.

REGISTER ADDRESS	REGISTER NAME	SYNONYM
MSDC + 0000h	MS/SD Memory Card Controller Configuration Register	MSDC_CFG
MSDC + 0004h	MS/SD Memory Card Controller Status Register	MSDC_STA
MSDC + 0008h	MS/SD Memory Card Controller Interrupt Register	MSDC_INT
MSDC + 000Ch	MS/SD Memory Card Controller Data Register	MSDC_DAT
MSDC + 00010h	MS/SD Memory Card Pin Status Register	MSDC_PS
MSDC + 00014h	MS/SD Memory Card Controller IO Control Register	MSDC_IOCON
MSDC + 0020h	SD Memory Card Controller Configuration Register	SDC_CFG
MSDC + 0024h	SD Memory Card Controller Command Register	SDC_CMD
MSDC + 0028h	SD Memory Card Controller Argument Register	SDC_ARG
MSDC + 002Ch	SD Memory Card Controller Status Register	SDC_STA
MSDC + 0030h	SD Memory Card Controller Response Register 0	SDC_RESP0
MSDC + 0034h	SD Memory Card Controller Response Register 1	SDC_RESP1
MSDC + 0038h	SD Memory Card Controller Response Register 2	SDC_RESP2
MSDC + 003Ch	SD Memory Card Controller Response Register 3	SDC_RESP3
MSDC + 0040h	SD Memory Card Controller Command Status Register	SDC_CMDSTA
MSDC + 0044h	SD Memory Card Controller Data Status Register	SDC_DATSTA
MSDC + 0048h	SD Memory Card Status Register	SDC_CSTA
MSDC + 004Ch	SD Memory Card IRQ Mask Register 0	SDC_IRQMASK0
MSDC + 0050h	SD Memory Card IRQ Mask Register 1	SDC_IRQMASK1
MSDC + 0054h	SDIO Configuration Register	SDIO_CFG
MSDC + 0058h	SDIO Status Register	SDIO_STA
MSDC + 0060h	Memory Stick Controller Configuration Register	MSC_CFG
MSDC + 0064h	Memory Stick Controller Command Register	MSC_CMD
MSDC + 0068h	Memory Stick Controller Auto Command Register	MSC_ACMD
MSDC + 006Ch	Memory Stick Controller Status Register	MSC_STA

Table 39 ***MS/SD Controller Register Map***

© 2009 MediaTek Inc.

2.23.3.1 Global Register Definitions

MSDC+0000h MS/SD Memory Card Controller Configuration Register MSDC_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFOTHD				PRCFG2		PRCFG1		PRCFG0		VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type	R/W				R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0001				01		01		10		0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC	RST	NOCRC		MSDC
Type	R/W								R/W	R/W	R/W	R/W	W	R/W		R/W
Reset	00000000								0	0	1	0	0	0		0

The register is for general configuration of the MS/SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

MSDC The register bit is used to configure the controller as the host of Memory Stick or as the host of SD/MMC Memory card. The default value is to configure the controller as the host of Memory Stick.

- **0** Configure the controller as the host of Memory Stick
- 1 Configure the controller as the host of SD/MMC Memory card

NOCRC CRC Disable. A ‘1’ indicates that data transfer without CRC is desired. For write data block, data will be transmitted without CRC. For read data block, CRC will not be checked. It is for testing purpose.

- **0** Data transfer with CRC is desired.
- **1** Data transfer without CRC is desired.

RST Software Reset. Writing a ‘1’ to the register bit will cause internal synchronous reset of MS/SD controller, but does not reset register settings.

- **0** Otherwise
- **1** Reset MS/SD controller

CLKSRC The register bit specifies which clock is used as the source clock of the memory card. If the MCU clock is used, the fastest clock rate the memory card is 122/4=30.5MHz. Instead, if the MCPLL clock is used, the fastest clock rate of the memory card is 91/2=45.5MHz.

- **0** Use MCU clock as the source clock of the memory card.
- **1** Use MCPLL clock as the source clock of the memory card.

STDBY Standby Mode. If the module is powered down, operating clock to the module will be stopped. At the same time, clock to card detection circuitry will also be stopped. If detection of memory card insertion and removal is desired, write ‘1’ to the register bit. If interrupt for detection of memory card insertion and removal is enabled, interrupt will take place whenever memory is inserted or removed.

- **0** Standby mode is disabled.
- **1** Standby mode is enabled.

RED Rise Edge Data. The register bit is used to determine that serial data input is latched at the falling edge or the rising edge of serial clock. The default setting is at the rising edge. If serial data has

worse timing, set the register bit to ‘1’. **When memory card has worse timing on return read data, set the register bit to ‘1’.**

0 Serial data input is latched at the rising edge of serial clock.

1 Serial data input is latched at the falling edge of serial clock.

SCLKON Serial Clock Always On. It is for debugging purpose.

0 Not to have serial clock always on.

1 To have serial clock always on.

SCLKF The register field controls clock frequency of serial clock on MS/SD bus. Denote clock frequency of MS/SD bus serial clock as f_{slave} and clock frequency of the MS/SD controller as f_{host} which is 104 or 52 MHz. Then the value of the register field is as follows. **Note that the allowable maximum frequency of f_{slave} is 26MHz. While changing clock rate, it needs “ 1T clock period before change + 1T clock period after change” for HW signal to re-synchronize.**

00000000b f_{slave} =(1/2) * f_{host}

00000001b f_{slave} = (1/(4*1)) * f_{host}

00000010b f_{slave} = (1/(4*2)) * f_{host}

00000011b f_{slave} = (1/(4*3))* f_{host}

…

00010000b f_{slave} = (1/(4*16))* f_{host}

…

11111111b f_{slave} = (1/(4*255)) * f_{host}

INTEN Interrupt Enable. Note that if interrupt capability is disabled then application software must poll the status of the register MSDC_STA to check for any interrupt request.

0 Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD/MMC Memory Card or Memory Stick.

1 Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD/MMC Memory Card or Memory Stick.

DMAEN DMA Enable. Note that if DMA capability is disabled then application software must poll the status of the register MSDC_STA for checking any data transfer request. If DMA is desired, the register bit must be set before command register is written.

0 DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD/MMC Memory Card or Memory Stick.

1 DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD/MMC Memory Card or Memory Stick.

PINEN Pin Interrupt Enable. The register bit is used to control if the pin for card detection is used as an interrupt source.

0 The pin for card detection is not used as an interrupt source.

1 The pin for card detection is used as an interrupt source.

DIRQEN Data Request Interrupt Enable. The register bit is used to control if data request is used as an interrupt source.

0 Data request is not used as an interrupt source.

1 Data request is used as an interrupt source.

RCDEN The register bit controls the output pin RCDEN that is used for card identification process when the controller is for SD/MMC Memory Card. Its output will control the pull down resistor on the system board to connect or disconnect with the signal CD/DAT3.

- **0** The output pin RCDEN will output logic low.
- **1** The output pin RCDEN will output logic high.

VDDPD The register bit controls the output pin VDDPD that is used for power saving. The output pin VDDPD will control power for memory card.

- **0** The output pin VDDPD will output logic low. The power for memory card will be turned off.
- **1** The output pin VDDPD will output logic high. The power for memory card will be turned on.

PRCFG0 Pull Up/Down Register Configuration for the pin WP. The default value is 10.

- **00** Pull up resistor and pull down resistor in the I/O pad of the pin WP are all disabled.
- **01** Pull down resistor in the I/O pad of the pin WP is enabled.
- **10** Pull up resistor in the I/O pad of the pin WP is enabled.
- **11** Use keeper of IO pad.

PRCFG1 Pull Up/Down Register Configuration for the pin CMD/BS. The default value is 0b01.

- **00** Pull up resistor and pull down resistor in the I/O pad of the pin CMD/BS are all disabled.
- **01** Pull down resistor in the I/O pad of the pin CMD/BS is enabled.
- **10** Pull up resistor in the I/O pad of the pin CMD/BS is enabled.
- **11** Use keeper of IO pad.

PRCFG2 Pull Up/Down Register Configuration for the pins DAT0, DAT1, DAT2, DAT3. The default value is 0b01.

- **00** Pull up resistor and pull down resistor in the I/O pads o the pins DAT0, DAT1, DAT2, DAT3. are all disabled.
- **01** Pull down resistor in the I/O pads of the pins DAT0, DAT1, DAT2, DAT3 and WP. is enabled.
- **10** Pull up resistor in the I/O pads of the pins DAT0, DAT1, DAT2, DAT3. is enabled.
- **11** Use keeper of IO pad.

FIFOTHD FIFO Threshold. The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are larger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are larger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001.

- **0000** Invalid.
- **0001** Threshold value is 1.
- **0010** Threshold value is 2.
- **…**
- **1000** Threshold value is 8.
- **others** Invalid

MSDC+0004h MS/SD Memory Card Controller Status Register MSDC_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY	FIFOCLR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W							RO				RO	RO	RO	RO
Reset	0	-							0000				0	0	0	0

The register contains the status of FIFO, interrupts and data requests.

BF The register bit indicates if FIFO in MS/SD controller is full.
- **0** FIFO in MS/SD controller is not full.
- **1** FIFO in MS/SD controller is full.

BE The register bit indicates if FIFO in MS/SD controller is empty.
- **0** FIFO in MS/SD controller is not empty.
- **1** FIFO in MS/SD controller is empty.

DRQ The register bit indicates if any data transfer is required. While any data transfer is required, the register bit still will be active even if the register bit DIRQEN in the register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. While the register bit DIRQEN in the register MSDC_CFG is disabled, the second method is used.
- **0** No DMA request exists.
- **1** DMA request exists.

INT The register bit indicates if any interrupt exists. While any interrupt exists, the register bit still will be active even if the register bit INTEN in the register MSDC_CFG is disabled. MS/SD controller can interrupt MCU by issuing interrupt request to Interrupt Controller, or software/application polls the register endlessly to check if any interrupt request exists in MS/SD controller. While the register bit INTEN in the register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that timeout error takes place. Software can read the status register to check if timeout error takes place without OS time tick support or data request is asserted. Note that the register bit will be cleared when reading the register MSDC_INT.
- **0** No interrupt request exists.
- **1** Interrupt request exists.

FIFOCNT FIFO Count. The register field shows how many valid entries are in FIFO.
- **0000** There is 0 valid entry in FIFO.
- **0001** There is 1 valid entry in FIFO.
- **0010** There are 2 valid entries in FIFO.
- **…**
- **1000** There are 8 valid entries in FIFO.
- **others** Invalid

FIFOCLR Clear FIFO. Writing ‘1’ to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.
- **0** No effect on FIFO.
- **1** Clear the content of FIFO clear and reset the status of FIFO controller.

BUSY Status of the controller. If the controller is in busy state, the register bit will be '1'. Otherwise '0'.

0 The controller is in busy state.

1 The controller is in idle state.

MSDC+0008h MS/SD Memory Card Controller Interrupt Register MSDC_INT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOIRQ	SDR1BIRQ	MSIFIRQ	SDMCIRQ	SDDATIRQ	SDCMDIRQ	PINIRQ	DIRQ
Type									RC	RC	RC	RC	RC	RC	RC	RC
Reset									0	0	0	0	0	0	0	0

The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, the register bit INTEN of the register MSDC_CFG is set to '0. It implies that software interrupt can be implemented by polling the register bit INT of the register MSDC_STA and this register. **However, if hardware interrupt is desired, remember to clear the register before setting the register bit INTEN of the register MSDC_CFG to '1'. Or undesired hardware interrupt arisen from previous interrupt status may take place.**

DIRQ Data Request Interrupt. The register bit indicates if any interrupt for data request exists. Whenever data request exists and data request as an interrupt source is enabled, i.e., the register bit DIRQEN in the register MSDC_CFG is set to '1', the register bit will be active. It will be reset when reading it. For software, data requests can be recognized by polling the register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers.

0 No Data Request Interrupt.

1 Data Request Interrupt occurs.

PINIRQ Pin Change Interrupt. The register bit indicates if any interrupt for memory card insertion/removal exists. Whenever memory card is inserted or removed and card detection interrupt is enabled, i.e., the register bit PINEN in the register MSDC_CFG is set to '1', the register bit will be set to '1'. It will be reset when the register is read.

0 Otherwise.

1 Card is inserted or removed.

SDCMDIRQ SD Bus CMD Interrupt. The register bit indicates if any interrupt for SD CMD line exists. Whenever interrupt for SD CMD line exists, i.e., any bit in the register SDC_CMDSTA is active, the register bit will be set to '1' if interrupt is enabled. It will be reset when the register is read.

0 No SD CMD line interrupt.

1 SD CMD line interrupt exists.

SDDATIRQ SD Bus DAT Interrupt. The register bit indicates if any interrupt for SD DAT line exists. Whenever interrupt for SD DAT line exists, i.e., any bit in the register SDC_ DATSTA is active, the register bit will be set to '1' if interrupt is enabled. It will be reset when the register is read.

0 No SD DAT line interrupt.

1 SD DAT line interrupt exists.

SDMCIRQ SD Memory Card Interrupt. The register bit indicates if any interrupt for SD Memory Card exists. Whenever interrupt for SD Memory Card exists, i.e., any bit in the register SDC_CSTA is active, the register bit will be set to '1' if interrupt is enabled. It will be reset when the register is read.

0 No SD Memory Card interrupt.
1 SD Memory Card interrupt exists.

MSIFIRQ MS Bus Interface Interrupt. The register bit indicates if any interrupt for MS Bus Interface exists. Whenever interrupt for MS Bus Interface exists, i.e., any bit in the register MSC_STA is active, the register bit will be set to '1' if interrupt is enabled. It will be reset when the register MSDC_STA or MSC_STA is read.

0 No MS Bus Interface interrupt.
1 MS Bus Interface interrupt exists.

SDR1BIRQ SD/MMC R1b Response Interrupt. The register bit will be active when a SD/MMC command with R1b response finishes and the DAT0 line has transition from busy to idle state. **Single block write commands with R1b response will cause the interrupt when the command completes no matter successfully or with CRC error. However, multi-block write commands with R1b response do not cause the interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause the interrupt. Single block read command with R1b response will cause the interrupt when the command completes but multi-block read commands do not. Note that STOP_TRANS commands (with R1b response) behind multi-block read commands will cause the interrupt.**

0 No interrupt for SD/MMC R1b response.
1 Interrupt for SD/MMC R1b response exists.

SDIOIRQ SDIO Interrupt. The register bit indicates if any interrupt for SDIO exists. Whenever interrupt for SDIO exists, i.e., the register bit will be set to '1' if interrupt is enabled. It will be reset when the register is read.

0 No SDIO interrupt.
1 SDIO Card interrupt exists.

MSDC+000Ch MS/SD Memory Card Controller Data Register MSDC_DAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DATA[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA[15:0]**															
Type	R/W															

The register is used to read/write data from/to FIFO inside MS/SD controller. Data access is in unit of 32 bits.

MSDC+0010h MS/SD Memory Card Pin Status Register MSDC_PS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**CMD**	**DAT**							
Type								RO	RO							
Reset								-	-							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CDDEBOUNCE**											**PINCHG**	**PIN0**	**POEN0**	**PIEN0**	**CDEN**

Type	RW											RC	RO	R/W	R/W	R/W
Reset	0000											0	1	0	0	0

The register is used for card detection. When the memory card controller is powered on, and the system is powered on, the power for the memory card is still off unless power has been supplied by the PMIC. Meanwhile, pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD/MMC.

For detecting card insertion, first pull up INS pin, and then enable card detection and input pin at the same time. After 32 cycles of controller clock, status of pin changes will emerge. For detecting card removal, just keep enabling card detection and input pin.

CDEN Card Detection Enable. The register bit is used to enable or disable card detection.
- **0** Card detection is disabled.
- **1** Card detection is enabled.

PIEN0 The register bit is used to control input pin for card detection.
- **0** Input pin for card detection is disabled.
- **1** Input pin for card detection is enabled.

POEN0 The register bit is used to control output of input pin for card detection.
- **0** Output of input pin for card detection is disabled.
- **1** Output of input pin for card detection is enabled.

PIN0 The register shows the value of input pin for card detection.
- **0** The value of input pin for card detection is logic low.
- **1** The value of input pin for card detection is logic high.

PINCHG Pin Change. The register bit indicates the status of card insertion/removal. If memory card is inserted or removed, the register bit will be set to '1' no matter pin change interrupt is enabled or not. It will be cleared when the register is read.
- **0** Otherwise.
- **1** Card is inserted or removed.

CDDEBOUNCE The register field specifies the time interval for card detection de-bounce. Its default value is 0. It means that de-bounce interval is 32 cycle time of 32KHz. The interval will extend one cycle time of 32KHz by increasing the counter by 1.

DAT Memory Card Data Lines.

CMD Memory Card Command Lines.

MSDC+0014h MS/SD Memory Card Controller IO Control Register MSDC_IOCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DLT**											**CRCDIS**	**CMDSEL**	**INTLH**		**DSW**
Type	R/W											R/W	R/W	R/W		R/W
Reset	00000010											0	0	0		0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**CMDRE**						**PRCFG3**		**SRCFG1**	**SRCFG0**	**ODCCFG1**			**ODCCFG0**		
Type	R/W						R/W		R/W	R/W	R/W			R/W		
Reset	0						10		1	1	000			011		

The register specifies **Output Driving Capability** and **Slew Rate** of IO pads for MSDC. The reset value is suggestion setting. If output driving capability of the pins DAT0, DAT1, DAT2 and DAT3 is too large, it's possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.

ODCCFG0 Output driving capability the pins CMD/BS and SCLK

- **000** 4mA
- **010** 8mA
- **100** 12mA
- **110** 16mA

ODCCFG1 Output driving capability the pins DAT0, DAT1, DAT2 and DAT3

- **000** 4mA
- **010** 8mA
- **100** 12mA
- **110** 16mA

SRCFG0 Output driving capability the pins CMD/BS and SCLK

- **0** Fast Slew Rate
- **1** Slow Slew Rate

SRCFG1 Output driving capability the pins DAT0, DAT1, DAT2 and DAT3

- **0** Fast Slew Rate
- **1** Slow Slew Rate

PRCFG3 Pull Up/Down Register Configuration for the pin INS. The default value is 10.

- **00** Pull up resistor and pull down resistor in the I/O pad of the pin INS are all disabled.
- **01** Pull down resistor in the I/O pad of the pin INS is enabled.
- **10** Pull up resistor in the I/O pad of the pin INS is enabled.
- **11** Use keeper of IO pad.

CMDRE The register bit is used to determine whether the host should latch response token (which is sent from card on CMD line) at rising edge or falling edge of serial clock.

- **0** Host latches response at rising edge of serial clock
- **1** Host latches response at falling edge of serial clock

DSW The register bit is used to determine whether the host should latch data with 1-T delay or not. For SD card, this bit is suggest to be 0. for MS/MSPRO cards, it is suggested to be 1. **Note that this field is added after MT6268 and MT6516. (TK6516 not support yet)**

- **0** Host latches the data with 1-T delay
- **1** Host latches the data without 1-T delay

INTLH This field is used to select the latch timing for SDIO multi-block read interrupt. **Note that this field is added after MT6268 and MT6516. (TK6516 not support yet)**

- **00** Host latches INT at the second backend clock after the end bit of current data block from card is received. (This is the default setting)
- **01** Host latches INT at the first backend clock after the end bit of current data block from card is received.

10 Host latches INT at the second backend clock after the end bit of current data block from card is received.

11 Host latches INT at the third backend clock after the end bit of current data block from card is received.

CMDSEL The register bit is used to determine whether the host should delay 1-T to latch response from card. **Note that this field is added after MT6268 and MT6516. (TK6516 not support yet)**

0 Host latches response without 1-T delay.

1 Host latches response with 1-T delay.

CRCDIS The register bit is used to switch-off the data CRC check for SD/MMC read data. **Note that this field is added after MT6268 and MT6516. (TK6516 not support yet)**

0 CRC Check is on.

1 CRC Check is off.

DLT Data Latch Timing. The register is used for SW to select the latch timing on data line.
Figure 3 illustrates the data line latch timing. sclk_out is the serial clock output to card. div_clk is the internal clock used for generating divided clock. The number "1 2 1 2" means the current sclk_out is divided from div_clk by a ratio of 2. data_in is the output data from card, and latched_data(r)/(f) is the rising/falling edge latched data inside the host (configured by RED in MSDC_CFG). In this example, SCLKF(in MSDC_CFG) is set to 8'b0 which means the division ratio is 2, and DLT is set to 1. Note that the value of DLT CANNOT be set as 0 and its value should not exceed the division ratio (in the example, the division ratio is 2). Also note that, the latching time will be one div_clk later than the indicated DLT value and the falling edge is always half div_clk ahead from rising edge. The default value of DLT is set to 8'b2.

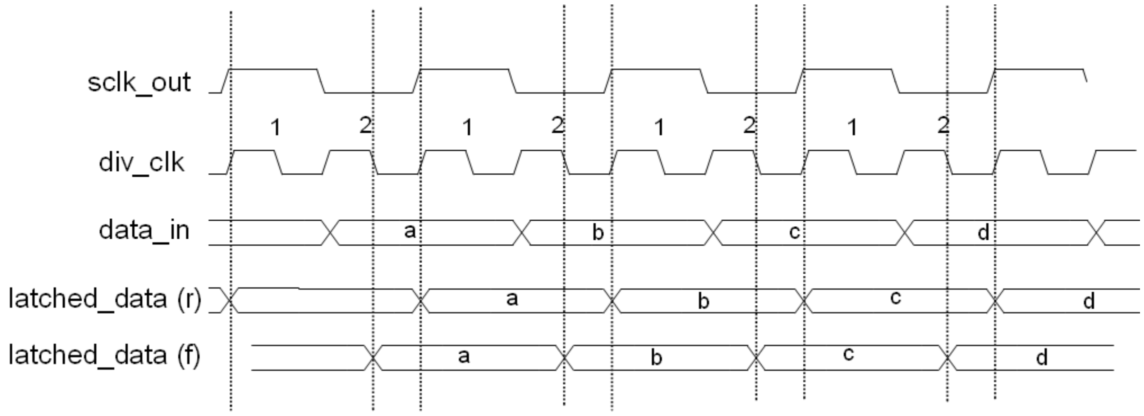


Figure 3 ***Illustration of data line latch timing***

2.23.3.2 SD Memory Card Controller Register Definitions

MSDC+0020h SD Memory Card Controller Configuration Register SDC_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTOC								WDOD				SDIO	MDLW8	MDLEN	SIEN
Type	R/W								R/W				R/W	R/W	R/W	R/W
Reset	00000000								0000				0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	R/W				R/W											
Reset	1000				00000000000											

The register is used for configuring the MS/SD Memory Card Controller when it is configured as the host of SD Memory Card. If the controller is configured as the host of Memory Stick, the contents of the register have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

BLKLEN It refers to Block Length. The register field is used to define the length of one block in unit of byte in a data transaction. The maximal value of block length is 2048 bytes.

000000000000 Reserved.

000000000001 Block length is 1 byte.

000000000010 Block length is 2 bytes.

…

011111111111 Block length is 2047 bytes.

100000000000 Block length is 2048 bytes.

BSYDLY The register field is only valid for the commands with R1b response. If the command has a response of R1b type, MS/SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if operations in SD/MMC Memory Card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, then the controller will abandon the detection.

0000 No extend.

0001 Extend one more serial clock cycle.

0010 Extend two more serial clock cycles.

…

1111 Extend fifteen more serial clock cycle.

SIEN Serial Interface Enable. It should be enabled as soon as possible before any command.

0 Serial interface for SD/MMC is disabled.

1 Serial interface for SD/MMC is enabled.

MDLW8 Eight Data Line Enable. The register works when MDLEN is enabled. The register can be enabled only when MultiMediaCard 4.0 is applied and detected by software application.

0 4-bit Data line is enabled.

1 8-bit Data line is enabled.

SDIO SDIO Enable.

0 SDIO mode is disabled

1 SDIO mode is enabled

MDLEN Multiple Data Line Enable. The register can be enabled only when SD Memory Card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an MultiMediaCard is applied. If an MultiMediaCard is applied and 4-bit data line is enabled, then 4 bits will be output every serial clock. Therefore, data integrity will fail.

0 4-bit Data line is disabled.

 © 2009 MediaTek Inc.

1 4-bit Data line is enabled.

WDOD Write Data Output Delay. The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock.

0000 No extend.
0001 Extend one more serial clock cycle.
0010 Extend two more serial clock cycles.
…
1111 Extend fifteen more serial clock cycle.

DTOC Data Timeout Counter. The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 65,536 serial clock. See the register field description of the register bit RDINT for reference.

00000000 Extend 65,536 more serial clock cycle.
00000001 Extend 65,536x2 more serial clock cycle.
00000010 Extend 65,536x3 more serial clock cycle.
…
11111111 Extend 65,536x 256 more serial clock cycle.

MSDC+0024h SD Memory Card Controller Command Register SDC_CMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**CMDFAIL**
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INTC**	**STOP**	**RW**	**DTYPE**		**IDRT**	**RSPTYP**			**BREAK**	**CMD**					
Type	R/W	R/W	R/W	R/W		R/W	R/W			R/W	R/W					
Reset	0	0	0	00		0	000			0	000000					

The register defines a SD Memory Card command and its attribute. Before MS/SD controller issues a transaction onto SD bus, application shall specify other relative setting such as argument for command. After application writes the register, MS/SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD/MMC bus run 128 cycles before issuing the command.

CMD SD Memory Card command. It is totally 6 bits.

BREAK Abort a pending MMC GO_IRQ_MODE command. It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response.

0 Other fields are valid.
1 Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.

RSPTYP The register field defines response type for the command. For commands with R1 and R1b response, the register SDC_CSTA (not SDC_STA) will update after response token is received. This

register SDC_CSTA contains the status of the SD/MMC and it will be used as response interrupt sources. Note that if CMD7 is used with all 0's RCA then RSPTYP must be "000". And the command "GO_TO_IDLE" also have RSPTYP='000'.

000 There is no response for the command. For instance, broadcast command without response and GO_INACTIVE_STATE command.

001 The command has R1 response. R1 response token is 48-bit.

010 The command has R2 response. R2 response token is 136-bit.

011 The command has R3 response. Even though R3 is 48-bit response, but it does not contain CRC checksum.

100 The command has R4 response. R4 response token is 48-bit. (Only for MMC)

101 The command has R5 response. R5 response token is 48-bit. (Only for MMC)

110 The command has R6 response. R6 response token is 48-bit.

111 The command has R1b response. If the command has a response of R1b type, MS/SD controller must monitor the data line 0 for card busy status from the bit time that is two or four serial clock cycles after the command end bit to check if operations in SD/MMC Memory Card have finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by four serial clock cycles. The second case is that the card is in idle state or under a scenario of receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit is valid only when the command has a response token.

IDRT Identification Response Time. The register bit indicates if the command has a response with N_{ID} (that is, 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to '1' for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD).

0 Otherwise.

1 The command has a response with N_{ID} response time.

DTYPE The register field defines data token type for the command.

00 No data token for the command

01 Single block transaction

10 Multiple block transaction. That is, the command is a multiple block read or write command.

11 Stream operation. It only shall be used when an MultiMediaCard is applied.

RW The register bit defines the command is a read command or write command. The register bit is valid only when the command will cause a transaction with data token.

0 The command is a read command.

1 The command is a write command.

STOP The register bit indicates if the command is a stop transmission command.

0 The command is not a stop transmission command.

1 The command is a stop transmission command.

INTC The register bit indicates if the command is GO_IRQ_STATE. If the command is GO_IRQ_STATE, the period between command token and response token will not be limited.

0 The command is not GO_IRQ_STATE.

1 The command is GO_IRQ_STATE.

MSDC+0028h SD Memory Card Controller Argument Register SDC_ARG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARG [31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARG [15:0]															
Type	R/W															

The register contains the argument of the SD/MMC Memory Card command.

MSDC+002Ch SD Memory Card Controller Status Register SDC_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WP											FEDATBUSY	FECMDBUSY	BEDATBUSY	BECMDBUSY	BESDCBUSY
Type	R											RO	RO	RO	RO	RO
Reset	-											0	0	0	0	0

The register contains various status of MS/SD controller as the controller is configured as the host of SD Memory Card.

BESDCBUSY The register field indicates if MS/SD controller is busy, that is, any transmission is going on CMD or DAT line on SD bus. This bit shows backend controller's SDC busy state. The busy state is sync from card clock domain to bus clock domain.

0 Backend MS/SD controller is idle.

1 Backend MS/SD controller is busy.

BECMDBUSY The register field indicates if any transmission is going on CMD line on SD bus. This bit shows backend controller's CMD busy state. The busy state is sync from card clock domain to bus clock domain.

0 Backend MS/SDC Controller gets the info that no transmission is going on CMD line on SD bus.

1 Backend MS/SDC Controller gets the info that there exists transmission going on CMD line on SD bus.

BEDATBUSY The register field indicates if any transmission is going on DAT line on SD bus.

0 Backend MS/SDC Controller gets the info that no transmission is going on DAT line on SD bus.

1 Backend MS/SDC Controller gets the info that there exists transmission going on DAT line on SD bus.

FECCMDBUSY The register field indicates if any transmission is going on CMD line on SD bus. This bit indicates directly the CMD line at card clock domain.

0 No transmission is going on CMD line on SD bus.

1 There exists transmission going on CMD line on SD bus.

FEDATBUSY The register field indicates if any transmission is going on DAT line on SD bus. This bit indicates directly the CMD line at card clock domain. **For those commands without data but still involving DAT line, the register bit is useless. For example, if an Erase command is issued,**

then checking if the register bit is '0' before issuing next command with data would not guarantee that the controller is idle. In this situation, use the register bit BESDCBUSY.

0 No transmission is going on DAT line on SD bus.

1 There exists transmission going on DAT line on SD bus.

WP It is used to detect the status of Write Protection Switch on SD Memory Card. The register bit shows the status of Write Protection Switch on SD Memory Card. There is no default reset value. The pin WP (Write Protection) is also only useful while the controller is configured for SD Memory Card.

1 Write Protection Switch ON. It means that memory card is desired to be write-protected.

0 Write Protection Switch OFF. It means that memory card is writable.

MSDC+0030h SD Memory Card Controller Response Register 0 SDC_RESP0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP [31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP [15:0]**															
Type	RO															

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

MSDC+0034h SD Memory Card Controller Response Register 1 SDC_RESP1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP [63:48]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP [47:32]**															
Type	RO															

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

MSDC+0038h SD Memory Card Controller Response Register 2 SDC_RESP2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP [95:80]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP [79:64]**															
Type	RO															

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

MSDC+003Ch SD Memory Card Controller Response Register 3 SDC_RESP3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP [127:112]**															
Type	RO															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP [111:96]**															
Type	RO															

The register contains parts of the last SD/MMC Memory Card bus response. The register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD/MMC Memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of response token is stored in the register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of response token is stored in the register field SDC_RESP0.

MSDC+0040h SD Memory Card Controller Command Status Register SDC_CMDSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**MMCIRQ**	**RSPCRCERR**	**CMDTO**	**CMDRDY**
Type													RC	RC	RC	RC
Reset													0	0	0	0

The register contains the status of MS/SD controller during command execution and that of MS/SD bus protocol after command execution when MS/SD controller is configured as the host of SD/MMC Memory Card. The register will also be used as interrupt sources. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

CMDRDY For command without response, the register bit will be ‘1’ once the command completes on SD/MMC bus. For command with response, the register bit will be ‘1’ whenever the command is issued onto SD/MMC bus and its corresponding response is received **without CRC error**.

- **0** Otherwise.
- **1** Command with/without response finish successfully without CRC error.

CMDTO Timeout on CMD detected. A ‘1’ indicates that MS/SD controller detected a timeout condition while waiting for a response on the CMD line.

- **0** Otherwise.
- **1** MS/SD controller detected a timeout condition while waiting for a response on the CMD line.

RSPCRCERR CRC error on CMD detected. A ‘1’ indicates that MS/SD controller detected a CRC error **after reading a response from the CMD line.**

- **0** Otherwise.
- **1** MS/SD controller detected a CRC error after reading a response from the CMD line.

MMCIRQ MMC requests an interrupt. A ‘1’ indicates that a MMC supporting command class 9 issued an interrupt request.

- **0** Otherwise.
- **1** A ‘1’ indicates that a MMC supporting command class 9 issued an interrupt request.

MSDC+0044h SD Memory Card Controller Data Status Register SDC_DATSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**DATCRCERR**	**DATTO**	**BLKDONE**
Type														RC	RC	RC
Reset														0	0	0

The register contains the status of MS/SD controller during data transfer on DAT line(s) when MS/SD controller is configured as the host of SD/MMC Memory Card. The register also will be used as interrupt sources. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

BLKDONE The register bit indicates the status of data block transfer.

- **0** Otherwise.
- 1 A data block was successfully transferred.

DATTO Timeout on DAT detected. A '1' indicates that MS/SD controller detected a timeout condition while waiting for data token on the DAT line.

- **0** Otherwise.
- **1** MS/SD controller detected a timeout condition while waiting for data token on the DAT line.

DATCRCERR CRC error on DAT detected. A '1' indicates that MS/SD controller detected a CRC error after reading a block of data from the DAT line or SD/MMC signaled a CRC error after writing a block of data to the DAT line.

- **0** Otherwise.
- **1** MS/SD controller detected a CRC error after reading a block of data from the DAT line or SD/MMC signaled a CRC error after writing a block of data to the DAT line.

MSDC+0048h SD Memory Card Status Register SDC_CSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CSTA [31:16]**															
Type	RC															
Reset	0000000000000000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CSTA [15:0]**															
Type	RC															
Reset	0000000000000000															

After commands with R1 and R1b response this register contains the status of the SD/MMC card and it will be used as response interrupt sources. In all register fields, logic high indicates error and logic low indicates no error. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

CSTA31 **OUT_OF_RANGE**. The command's argument was out of the allowed range for this card.

CSTA30 **ADDRESS_ERROR**. A misaligned address that did not match the block length was used in the command.

CSTA29 **BLOCK_LEN_ERROR**. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.

CSTA28 **ERASE_SEQ_ERROR**. An error in the sequence of erase commands occurred.

CSTA27 **ERASE_PARAM**. An invalid selection of write-blocks for erase occurred.

CSTA26 **WP_VIOLATION**. Attempt to program a write-protected block.

CSTA25 Reserved. Return zero.

CSTA24 **LOCK_UNLOCK_FAILED**. Set when a sequence or password error has been detected in lock/unlock card command or if there was an attempt to access a locked card.

CSTA23 **COM_CRC_ERROR**. The CRC check of the previous command failed.

CSTA22 **ILLEGAL_COMMAND**. Command not legal for the card state.

CSTA21 **CARD_ECC_FAILED**. Card internal ECC was applied but failed to correct the data.

CSTA20 **CC_ERROR**. Internal card controller error.

CSTA19 **ERROR**. A general or an unknown error occurred during the operation.

CSTA18 **UNDERRUN**. The card could not sustain data transfer in stream read mode.

CSTA17 **OVERRUN**. The card could not sustain data programming in stream write mode.

CSTA16 **CID/CSD_OVERWRITE**. It can be either one of the following errors: 1. The CID register has been already written and cannot be overwritten 2. The read only section of the CSD does not match the card. 3. An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.

CSTA[15:4] Reserved. Return zero.

CSTA3 **AKE_SEQ_ERROR**. Error in the sequence of authentication process

CSTA[2:0] Reserved. Return zero.

MSDC+004Ch SD Memory Card IRQ Mask Register 0 — SDC_IRQMASK 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQMASK [31:16]**															
Type	R/W															
Reset	0000000000000000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQMASK [15:0]**															
Type	R/W															
Reset	0000000000000000															

The register contains parts of SD Memory Card Interrupt Mask Register. See the register description of the register SDC_IRQMASK1 for reference. The register will mask interrupt sources from the register SDC_CMDSTA and SDC_DATSTA. IRQMASK[15:0] is for SDC_CMDSTA and IRQMASK[31:16] for SDC_DATSTA. A ‘1’ in some bit of the register will mask the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is ‘1’ then interrupt source from the register field CMDRDY of the register SDC_ CMDSTA will be masked. A ‘0’ in some bit will not cause interrupt mask on the corresponding interrupt source from the register SDC_CMDSTA and SDC_DATSTA.

 © 2009 MediaTek Inc.

MSDC+0050h SD Memory Card IRQ Mask Register 1 SDC_IRQMASK1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:48]															
Type	R/W															
Reset	0000000000000000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [47:32]															
Type	R/W															
Reset	0000000000000000															

The register contains parts of SD Memory Card Interrupt Mask Register. The registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD Memory Card Interrupt Mask Register. The register will mask interrupt sources from the register SDC_CSTA. A ‘1’ in some bit of the register will mask the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is ‘1’ then interrupt source from the register field OUT_OF_RANGE of the register SDC_ CSTA will be masked. A ‘0’ in some bit will not cause interrupt mask on the corresponding interrupt source from the register SDC_ CSTA.

MSDC+0054h SDIO Configuration Register SDIO_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														DSBSEL	INTSEL	INTEN
Type														R/W	R/W	R/W
Reset														0	0	0

The register is used to configure functionality for SDIO.

INTEN Interrupt enable for SDIO.

- **0** Disable
- **1** Enable

INTSEL Interrupt Signal Selection

- **0** Use data line 1 as interrupt signal
- **1** Use data line 5 as interrupt signal

DSBSEL Data Block Start Bit Selection.

- **0** Use data line 0 as start bit of data block and other data lines are ignored.
- **1** Start bit of a data block is received only when data line 0-3 all become low.

MSDC+0058h SDIO Status Register SDIO_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name															SDIOIRQ
Type															RO
Reset															0

SDIOIRQ SDIO Interrupt. The register bit indicates if any interrupt for SDIO exists. Whenever interrupt for SDIO exists, i.e., the register bit will be set to '1' if interrupt is enabled.

- **0** No SDIO interrupt.
- **1** SDIO Card interrupt exists.

Memory Stick Controller Register Definitions

MSDC+0060h Memory Stick Controller Configuration Register MSC_CFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PMODE	PRED											BUSYCNT			SIEN
Type	R/W	R/W											R/W			R/W
Reset	0	0											101			0

The register is used for Memory Stick Controller Configuration when MS/SD controller is configured as the host of Memory Stick.

SIEN Serial Interface Enable. It should be enabled as soon as possible before any command.

- **0** Serial interface for Memory Stick is disabled.
- **1** Serial interface for Memory Stick is enabled.

BUSYCNT RDY timeout setting in unit of serial clock cycle. The register field is set to the maximum BUSY timeout time (set value x 4 +2) to wait until the RDY signal is output from the card. RDY timeout error detection is not performed when BUSYCNT is set to 0. The initial value is 0x5. That is, BUSY signal exceeding 5x4+2=22 serial clock cycles causes a RDY timeout error.

- **000** Not detect RDY timeout
- **001** BUSY signal exceeding 1x4+2=6 serial clock cycles causes a RDY timeout error.
- **010** BUSY signal exceeding 2x4+2=10 serial clock cycles causes a RDY timeout error.
- **…**
- **111** BUSY signal exceeding 7x4+2=30 serial clock cycles causes a RDY timeout error.

PRED Parallel Mode Rising Edge Data. The register field is only valid in parallel mode, that is, MSPRO mode. In parallel mode, data must be driven and latched at the falling edge of serial clock on MS bus. In order to mitigate hold time issue, the register can be set to '1' such that write data is driven by MSDC at the rising edge of serial clock on MS bus.

- **0** Write data is driven by MSDC at the falling edge of serial clock on MS bus.
- **1** Write data is driven by MSDC at the rising edge of serial clock on MS bus.

PMODE Memory Stick PRO Mode.

- **0** Use Memory Stick serial mode.
- **1** Use Memory Stick parallel mode.

MSDC+0064h Memory Stick Controller Command Register MSC_CMD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	PID						DATASIZE									
Type	R/W						R/W									
Reset	0000						0000000000									

The register is used for issuing a transaction onto MS bus. Transaction on MS bus is started by writing to the register MSC_CMD. The direction of data transfer, that is, read or write transaction, is extracted from the register field PID. 16-bit CRC will be transferred for a write transaction even if the register field DATASIZE is programmed as zero under the condition where the register field NOCRC in the register MSDC_CFG is ‘0’. If the register field NOCRC in the register MSDC_CFG is ‘1’ and the register field DATASIZE is programmed as zero, then writing to the register MSC_CMD will not induce transaction on MS bus. The same applies for when the register field RDY in the register MSC_STA is ‘0’.

DATASIZE Data size in unit of byte for the current transaction.

- **0000000000** Data size is 0 byte.
- **0000000001** Data size is one byte.
- **0000000010** Data size is two bytes.
- **…**
- **0111111111** Data size is 511 bytes.
- **1000000000** Data size is 512 bytes.

PID Protocol ID. It is used to derive Transfer Protocol Code (TPC). The TPC can be derived by cascading PID and its reverse version. For example, if PID is 0x1, then TPC is 0x1e, that is, 0b0001 cascades 0b1110. In addition, the direction of the bus transaction can be determined from the register bit 15, that is, PID[3].

MSDC+0068h Memory Stick Controller Auto Command Register MSC_ACMD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	APID					ADATASIZE										ACEN
Type	R/W					R/W										R/W
Reset	0111					0000000001										0

The register is used for issuing a transaction onto MS bus automatically after the MS command defined in MSC_CMD completed on MS bus. Auto Command is a function used to automatically execute a command like GET_INT or READ_REG for checking status after SET_CMD ends. If auto command is enabled, the command set in the register will be executed once the INT signal on MS bus is detected. After auto command is issued onto MS bus, the register bit ACEN will become disabled automatically. Note that if auto command is enabled then the register bit RDY in the register MSC_STA caused by the command defined in MSC_CMD will be suppressed until auto command completes. Note that the register field ADATASIZE cannot be set to zero, or the result will be unpredictable.

ACEN Auto Command Enable.

- **0** Auto Command is disabled.
- **1** Auto Command is enabled.

ADATASIZE Data size in unit of byte for Auto Command. Initial value is 0x01.

- **0000000000** Data size is 0 byte.
- **0000000001** Data size is one byte.

0000000010 Data size is two bytes.

…

0111111111 Data size is 511 bytes.

1000000000 Data size is 512 bytes.

APID Auto Command Protocol ID. It is used to derive Transfer Protocol Code (TPC). Initial value is GSET_INT(0x7).

MSDC+006Ch Memory Stick Controller Status Register MSC_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDNK	BREQ	ERR	CED								HSRDY	CRCER	TOER	SIF	RDY
Type	R	R	R	R								RO	RO	RO	RO	RO
Reset	0	0	0	0								0	0	0	0	1

The register contains various status of Memory Stick Controller, that is, MS/SD controller is configured as Memory Stick Controller. These statuses can be used as interrupt sources. Reading the register will NOT clear it. The register will be cleared whenever a new command is written to the register MSC_CMD.

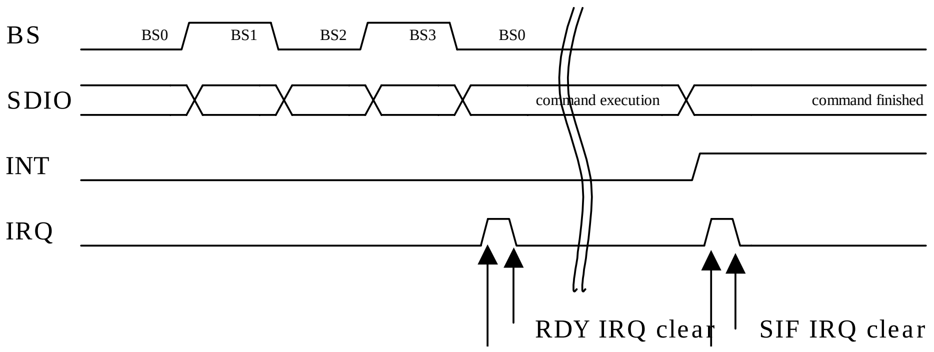
RDY The register bit indicates the status of transaction on MS bus. The register bit will be cleared when writing to the command register MSC_CMD.

0 Otherwise.

1 A transaction on MS bus is ended.

SIF The register bit indicates the status of serial interface. If an interrupt is active on MS bus, the register bit will be active. Note the difference between the signal RDY and SIF. When parallel mode is enabled, the signal SIF will be active whenever any of the signal CED, ERR, BREQ and CMDNK is active. **In order to separate interrupts caused by the signals RDY and SIF, the register bit SIF will not become active until the register MSDC_INT is read once. That is, the sequence for detecting the register bit SIF by polling is as follows:**

1. Detect the register bit RDY of the register MSC_STA
2. Read the register MSDC_INT
3. Detect the register bit SIF of the register MSC_STA



0 Otherwise.

1 An interrupt is active on MS bus

TOER The register bit indicates if a BUSY signal timeout error takes place. When timeout error occurs, the signal BS will become logic low '0'. The register bit will be cleared when writing to the command register MSC_CMD.

- **0** No timeout error.
- **1** A BUSY signal timeout error takes place. The register bit RDY will also be active.

CRCER The register bit indicates if a CRC error occurs while receiving read data. The register bit will be cleared when writing to the command register MSC_CMD.

- **0** Otherwise.
- **1** A CRC error occurs while receiving read data. The register bit RDY will also be active.

HSRDY The register bit indicates the status of handshaking on MS bus. The register bit will be cleared when writing to the command register MSC_CMD.

- **0** Otherwise.
- **1** A Memory Stick card responds to a TPC by RDY.

CED The register bit is only valid when parallel mode is enabled. In fact, it's value is from DAT[0] when serial interface interrupt takes place. See Format Specification version 2.0 of Memory Stick Standard (Memory Stick PRO) for more details.

- **0** Command does not terminate.
- **1** Command terminates normally or abnormally.

ERR The register bit is only valid when parallel mode is enabled. In fact, it's value is from DAT[1] when serial interface interrupt takes place. See Format Specification version 2.0 of Memory Stick Standard (Memory Stick PRO) for more details.

- **0** Otherwise.
- **1** Indicate memory access error during memory access command.

BREQ The register bit is only valid when parallel mode is enabled. In fact, it's value is from DAT[2] when serial interface interrupt takes place. See Format Specification version 2.0 of Memory Stick Standard (Memory Stick PRO) for more details.

- **0** Otherwise.
- **1** Indicate request for data.

CMDNK The register bit is only valid when parallel mode is enabled. In fact, it's value is from DAT[3] when serial interface interrupt takes place. See Format Specification version 2.0 of Memory Stick Standard (Memory Stick PRO) for more details.

- **0** Otherwise
- **1** Indicate non-recognized command.

2.23.4 Application Notes

2.23.4.1 Initialization Procedures After Power On

Disable power down control for MSDC module

Remember to power on MSDC module before starting any operation to it.

2.23.4.2 Card Detection Procedures

The pseudo code is as follows:

MSDC_CFG.PRCFG0 = 2'b10

 © 2009 MediaTek Inc.

```
MSDC_PS = 2'b11
MSDC_CFG.VDDPD = 1
if(MSDC_PS.PINCHG) { // card is inserted
        . . .
}
```

The pseudo code segment perform the following tasks:

1. First pull up CD/DAT3 (INS) pin.
2. Enable card detection and input pin at the same time.
3. Turn on power for memory card.
4. Detect insertion of memory card.

2.23.4.3 Notes on Commands

For MS, check if MSC_STA.RDY is '1' before issuing any command.

For SD/MMC, if the command desired to be issued involves data line, for example, commands with data transfer or R1b response, check if SDC_STA.SDCBUSY is '0' before issuing. If the command desired to be issued does not involve data line, only check if SDC_STA.CMDBUSY is '0' before issuing.

2.23.4.4 Notes on Data Transfer

- For SD/MMC, if multiple-block-write command is issued then only issue STOP_TRANS command inter-blocks instead of intra-blocks.
- Once SW decides to issue STOP_TRANS commands, no more data transfer from or to the controller.

2.23.4.5 Notes on Frequency Change

Before changing the frequency of serial clock on MS/SD/MMC bus, it is necessary to disable serial interface of the controller. That is, set the register bit SIEN of the register SDC_CFG to '0' for SD/MMC controller, and set the register bit SIEN of the register MSC_CFG to '0' for Memory Stick controller. Serial interface of the controller needs to be enabled again before starting any operation to the memory card.

2.23.4.6 Notes on Response Timeout

If a read command doest not receive response, that is, it terminates with a timeout, then register SDC_DATSTA needs to be cleared by reading it. The register bit "DATTO" should be active. However, it may take a while before the register bit becomes active. The alternative is to send the STOP_TRANS command. However, this method will receive response with illegal-command information. Also, remember to check if the register bit SDC_STA.CMDBUSY is active before issuing the STOP_TRANS command. The procedure is as follows:

1. Read command => response time out
2. Issue STOP_TRANS command => Get Response
3. Read register SDC_DATSTA to clear it

© 2009 MediaTek Inc.

2.23.4.7 Source or Destination Address is not word-aligned

It is possible that the source address is not word-aligned when data move from memory to MSDC. Similarly, destination address may be not word-aligned when data move from MSDC to memory. This can be solved by setting DMA byte-to-word functionality.

1. DMAn_CON.SIZE=0
2. DMAn_CON.BTW=1
3. DMAn_CON.BURST=2 (or 4)
4. DMAn_COUNT=byte number instead of word number
5. fifo threshold setting must be 1 (or 2), depending on DMAn_CON.BURST

Note n=4 ~ 11

2.23.4.8 Miscellaneous notes

- Siemens MMC card: When a write command is issued and followed by a STOP_TRANS command, Siemens MMC card will de-assert busy status even though flash programming has not yet finished. Software must use “Get Status” command to make sure that flash programming finishes.

2.24 NAND FLASH Interface

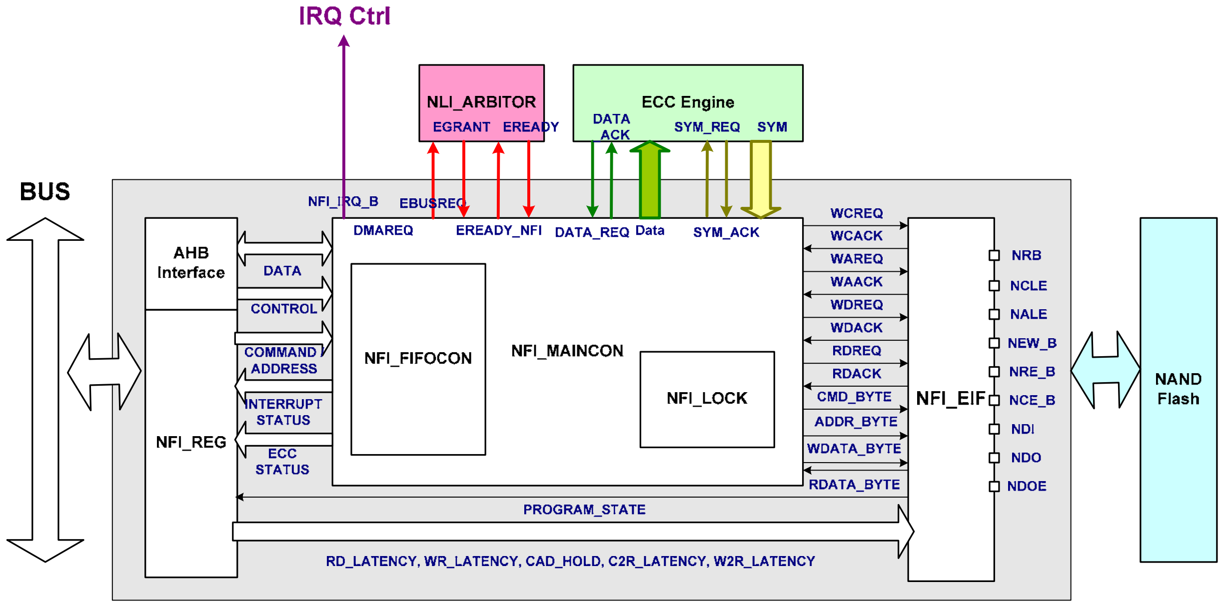


Figure 21 ***Block Diagram of NAND Flash Interface***

MT6516 provides NAND flash interface (for SLC & MLC).
The NAND FLASH interface support features as follows:

- ECC (BCH code) acceleration capable of 12 bit error correction. (with ECC engine)
- Programmable page size and spare size

- Programmable FDM data size and protected FDM data size.
- Word/byte access through APB bus.
- AHB for massive data transfer.
- Latch sensitive interrupt to indicate ready state for read, program, erase operation.
- Programmable wait states, command/address setup and hold time, read enable hold time, and write enable recovery time.
- Support 2 chip select for NAND flash parts.
- Support 8/16 bits I/O interface.

The NFI and ECC engine (in NFI mode) can automatically generate ECC syndrome bits when programming or reading the device. If the user approves the way it stores the syndrome bits in the spare area for each page, the HW_ECC mode can be used. Otherwise, the user can prepare the data (may contains operating system information or ECC syndrome bits) for the spare area with another arrangement. In the former case, the NFI and ECC engine (in NFI mode) can check the syndrome bits when reading from the device. The ECC module features the BCH code, which is capable of correcting 4/6/12 bit errors within one sector.

2.24.1 General description

2.24.1.1 Input and Output Interface

2.24.1.1.1 ECC Engine

Signal Name	Direction	Function
Auto_FMT_EN	Output	NFI Automatic append FDM data and encode or decode by HW ECC engine
B16EN	Output	8/16 bits I/O Data
Cur_Sec_num	Output	Read or write sector number as using AHB
ECC_str	Output	ECC Start trigger for each sector
Sec_Str_Addr	Output	Start Address of AHB for each sector
ECC_rdy	Input	ECC data request
ECC_valid	Output	ECC data valid ack
Data[15:0]	Output	Data for ECC
Par_req	Output	Parity Request to ECC
Par_rdy	Input	Parity ack from ECC
Parity[15:0]	Input	Parity

Table 40

2.24.1.1.2 NLI Arbiter and NAND Flash Device

Signal Name	Direction	Function
Egrant	Input	NLI bus grant indication
Eready	Input	NLI bus ready indication

Eready_nfi	Output	NFI ready indication for NLI bus ready
Ebusreq	Output	NLI bus request
NDI	Input	NAND Data Input
NDO	Output	NAND Data Output
NDOE	Output	NANA Data Output Enable
NRB	Output	NAND Flash Ready/Busy
NCE[1:0]	Output	NAND Flash Chip Enable
NWR	Output	NAND Flash Write Enable
NRE	Output	NAND Flash Read Enable
NCLE	Output	NAND Flash Command Latch Enable
NALE	Output	NAND Flash Address Latch Enable

Table 2

2.24.1.1.3 MCU

Signal Name	Direction	Function
nfi_irq_b	Output	Interrupt for MCU

Table 41

2.24.1.1.4 AHB Arbiter

Signal Name	Direction	Function
write	Output	Read or write
Address	Output	AHB Address
wdata[31:0]	Output	Write Data bus
rdata[31:0]	Input	Input Data bus
Byte_data	Output	Byte data or 4-byte data
AHB_req	Output	AHB request
AHB_ack	Input	AHB ack

Table 2

2.24.1.1.5 APB

Signal Name	Direction	Function
paddr[15:0]	Input	16 LSB of the ARM Address Bus
pwrite	Input	Write Control
penable	Input	Enable Interface
psel	Input	Transmitter Interface Select
pwdata[31:0]	Input	32 Bits of the ARM Writer Data Bus
prdata[31:0]	Output	32 Bits of the ARM Read Data Bus

© 2009 MediaTek Inc.

Table 3

2.24.2 Registers Memory Map

2.24.2.1 Registers Memory Map

Software responsibility and controllable functions

Register Address	Acronym	Register Function
NFI +0000h	NFI_CNFG	NFI Configuration
NFI +0004h	NFI_PAGEFMT	NFI Page Format Control
NFI +0008h	NFI_CON	NFI Control
NFI +000Ch	NFI_ACCCON	NAND Flash Access Control
NFI +0010h	NFI_INTR_EN	NFI Interrupt Enable
NFI +0014h	NFI_INTR	NFI Interrupt Status
NFI +0020h	NFI_CMD	NFI Command
NFI +0030h	NFI_ADDRNOB	NFI Address Length
NFI +0034h	NFI_COLADDR	NFI Column Address
NFI +0038h	NFI_ROWADDR	NFI Row Address
NFI +0040h	NFI_STRDATA	NFI Data Transfer Start Trigger
NFI +0050h	NFI_DATAW	Write Data Buffer
NFI +0054h	NFI_DATAWB	Write Data Buffer for Byte Access
NFI +0054h	NFI_DATAR	Read Data Buffer
NFI +005Ch	NFI_DATARB	Read Data Buffer for Byte Access
NFI +0060h	NFI_STA	NFI Status
NFI +0064h	NFI_FIFOSTA	NFI FIFO Status
NFI +0068h	NFI_LOCKSTA	NFI Lock Status
NFI +0070h	NFI_ADDRCNTR	NAND Flash Page Address Counter
NFI +0080h	NFI_STRADDR	AHB Start Address
NFI +0084h	NFI_BYTELEN	AHB Byte Length
I/O Pin Control		
NFI +0090h	NFI_CSEL	NAND Flash Device Select
NFI +0094h	NFI_IOCON	NFI IO Control
FDM Data Content		

 © 2009 MediaTek Inc.

NFI +00A0h	NFI_FDM0L	NFI Least FDM Data for Sector 0
NFI +00A4h	NFI_FDM0M	NFI Significant FDM Data for Sector 0
NFI +00A8h	NFI_FDM1L	NFI Least FDM Data for Sector 1
NFI +00ACh	NFI_FDM1M	NFI Significant FDM Data for Sector 1
NFI +00B0h	NFI_FDM2L	NFI Least FDM Data for Sector 2
NFI +00B4h	NFI_FDM2M	NFI Significant FDM Data for Sector 2
NFI +00B8h	NFI_FDM3L	NFI Least FDM Data for Sector 3
NFI +00BCh	NFI_FDM3M	NFI Significant FDM Data for Sector 3
NFI +00C0h	NFI_FDM4L	NFI Least FDM Data for Sector 4
NFI +00C4h	NFI_FDM4M	NFI Significant FDM Data for Sector 4
NFI +00C8h	NFI_FDM5L	NFI Least FDM Data for Sector 5
NFI +00CCh	NFI_FDM5M	NFI Significant FDM Data for Sector 5
NFI +00D0h	NFI_FDM6L	NFI Least FDM Data for Sector 6
NFI +00D4h	NFI_FDM6M	NFI Significant FDM Data for Sector 6
NFI +00D8h	NFI_FDM7L	NFI Least FDM Data for Sector 7
NFI +00DCh	NFI_FDM7M	NFI Significant FDM Data for Sector 7
	Flash Lock	
NFI +0100h	NFI_LOCK	NFI Lock Enable
NFI +0104h	NFI_LOCKCON	NFI Lock Contorl
NFI +0108h	NFI_LOCKANOB	NFI Address Format for Lock
NFI +0110h	NFI_LOCK00ADD	Row Start Address for Lock Set 00
NFI +0114h	NFI_LOCK00FMT	Row Address Format for Lock Set 00
NFI +0118h	NFI_LOCK01ADD	Row Start Address for Lock Set 01
NFI +011Ch	NFI_LOCK01FMT	Row Address Format for Lock Set 01
NFI +0120h	NFI_LOCK02ADD	Row Start Address for Lock Set 02
NFI +0124h	NFI_LOCK02FMT	Row Address Format for Lock Set 02
NFI +0128h	NFI_LOCK03ADD	Row Start Address for Lock Set 03
NFI +012Ch	NFI_LOCK03FMT	Row Address Format for Lock Set 03
NFI +0130h	NFI_LOCK04ADD	Row Start Address for Lock Set 04
NFI +0134h	NFI_LOCK04FMT	Row Address Format for Lock Set 04
NFI +0138h	NFI_LOCK05ADD	Row Start Address for Lock Set 05
NFI +013Ch	NFI_LOCK05FMT	Row Address Format for Lock Set 05
NFI +0140h	NFI_LOCK06ADD	Row Start Address for Lock Set 06
NFI +0144h	NFI_LOCK06FMT	Row Address Format for Lock Set 06
NFI +0148h	NFI_LOCK07ADD	Row Start Address for Lock Set 07

© 2009 MediaTek Inc.

NFI +014Ch	NFI_LOCK07FMT	Row Address Format for Lock Set 07
NFI +0150h	NFI_LOCK08ADD	Row Start Address for Lock Set 08
NFI +0154h	NFI_LOCK08FMT	Row Address Format for Lock Set 08
NFI +0158h	NFI_LOCK09ADD	Row Start Address for Lock Set 09
NFI +015Ch	NFI_LOCK09FMT	Row Address Format for Lock Set 09
NFI +0160h	NFI_LOCK10ADD	Row Start Address for Lock Set 10
NFI +0164h	NFI_LOCK10FMT	Row Address Format for Lock Set 10
NFI +0168h	NFI_LOCK11ADD	Row Start Address for Lock Set 11
NFI +016Ch	NFI_LOCK11FMT	Row Address Format for Lock Set 11
NFI +0170h	NFI_LOCK12ADD	Row Start Address for Lock Set 12
NFI +0174h	NFI_LOCK12FMT	Row Address Format for Lock Set 12
NFI +0178h	NFI_LOCK13ADD	Row Start Address for Lock Set 13
NFI +017Ch	NFI_LOCK13FMT	Row Address Format for Lock Set 13
NFI +0180h	NFI_LOCK14ADD	Row Start Address for Lock Set 14
NFI +0184h	NFI_LOCK14FMT	Row Address Format for Lock Set 14
NFI +0188h	NFI_LOCK15ADD	Row Start Address for Lock Set 15
NFI +018Ch	NFI_LOCK15FMT	Row Address Format for Lock Set 15
	Debug Register	
NFI +0190h	NFI_FIFODATA0	NFI FIFO Data 0
NFI +0194h	NFI_FIFODATA1	NFI FIFO Data 1
NFI +0198h	NFI_FIFODATA2	NFI FIFO Data 2
NFI +019Ch	NFI_FIFODATA3	NFI FIFO Data 3

Table 5 ***Registers Memory Map Table***

2.24.2.2 Register definition

NFI +0000h NFI Configuration NFI_CNFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**OP_MODE**					**AUTO_FMT_EN**	**HW_ECC_EN**		**BYTE_RW**					**READ_MODE**	**AHB_MODE**
Type		R/W					R/W	R/W		R/W					R/W	R/W
Reset		0					0	0		0					0	0

The register controls the NFI functions.

For all enable fields, Setting to be logic-1 represents enabled, while 0 represents disabled.

AHB_MODE This field is used to control the Operation mode.

0 MCU mode. All data (include read or write) move by MCU through APB access.

1 AHB mode. All data (include read or write) move by HW automation through AHB bus.

READ_EN This field is used to control the activity of read or write transfer.

0 write operation of AHB or MCU.

1 read operation of AHB or MCU.

BYTE_RW **Enable byte access**. The valid bytes read from NFI_DATAR and NFI_DATAW is only DR0 and DW0 if BYTE_RW is enabled.

HW_ECC_EN This field is used to enable encoding or decoding operation of HW ECC engine.

AUTO_FMT_EN Automatic HW ECC encode or decode enable.
If enabled, the ECC parity from HW ECC engine and FDM data from Register are written automatically to the spare area. If disable, the spare data all comes from MCU or AHB as main area data.

AUTO_FMT_EN	HW_ECC_EN	NFI Function
0	0	Data, FDM, Parity all come from MCU or AHB ECC Interface turn off
0	1	Data, FDM, Parity all come from MCU or AHB ECC Interface turn on
1	0	Data comes from MCU or AHB, FDM comes from Register and Parity forces 0xff ECC Interface turn off
1	1	Data comes from MCU or AHB, FDM comes from Register and Parity comes from ECC engine ECC Interface turn on

OP_MODE The field control the operating process flow of FSM for NFI..

000 Idle state.

001 Read Process. Recommend for basic read operation.

010 Single Read Process. Recommend for read id and read status.

011 Program Process. Recommend for basic program operation.

100 Erase Process. Recommend for basic erase operation.

101 Reset Process. Recommend for basic reset operation.

110 Custom Process. Recommend for all advance operation.

Others Reserved

NFI +0004h NFI Page Format Control Register NFI_PAGEFMT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FDM_ECC_NUM				FDM_NUM						SPARE_SIZE		DBYTE_EN		PAGE_SIZE	
Type	R/W				R/W						R/W		R/W		R/W	
Reset	0				0						0		0		0	

This register manages the page format of the device. It includes the bus width selection, the page size, the associated address format, and the spare format.

PAGE_SIZE Page Size. The field specifies the size of one page for the device. Some most widely used page size are supported.

0 The page size is 512 bytes (including 512 bytes data area and (spare_size*1) bytes spare area).

1 The page size is 2k bytes (including 2048 bytes data area and (spare_size*4) bytes spare area).

2 The page size is 4k bytes (including 4096 bytes data area and (spare_size*8) bytes spare area).

3 Reserved.

DBYTE_EN 16 bits I/O bus interface enable.

SPARE_SIZE Spare size per 512 bytes main area. 512 byte main area with spare size means one sector.

0 16 bytes.

1 26 bytes.

2 27 bytes.

3 28 bytes.

FDM_NUM The FDM data number for each spare area. The valid number of bytes are from 0 to 8.

FDM_ECC_NUM The number of each FDM data for HW ECC protection. The valid number of bytes ranges are from 0 to 8.

NFI +0008h NFI Operation Control Register NFI_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEC_NUM**						**BWR**	**BRD**	**NOB**			**SRD**			**NFI_RST**	**FIFO_FLUSH**
Type	R/W						R/W	R/W	W/R			WO			WO	WO
Reset	0						0	0	0			0			0	0

This is recommended to reset the state machine, data FIFO and flush the data FIFO before starting a new command

NFI_RST Reset the state machine, data FIFO (0x0000) and FDM data (0xffff) .

FIFO_FLUSH Flush the data FIFO.

This following register controls the burst mode and the single of the data access. In burst mode, the core supposes there are one or more than one page of data to be accessed. On the contrary, in single mode, the core supposes there are only less than 4 bytes of data to be accessed.

SRD Setting to be logic-1 initializes the one-shot data read operation. It's mainly used for read ID and read status command, which requires no more than 4 read cycles to retrieve data from the device. It used when FIFO is empty or after reset nficore

NOB The field represents the number of bytes to be retrieved from the device in single mode, and the number of bytes per APB transaction in both single and burst mode. If device is 16-bit IO, the read bytes number will double

0 Read 8 bytes from the device. (16 byte for 16-bit IO)

1 Read 1 byte from the device. (2 byte for 16-bit IO)

2 Read 2 bytes from the device. (4 byte for 16-bit IO)

3 Read 3 bytes from the device. (6 byte for 16-bit IO)

4 Read 4 bytes from the device. (8 byte for 16-bit IO)
5 Read 5 byte from the device. (10 byte for 16-bit IO)
6 Read 6 bytes from the device. (12 byte for 16-bit IO)
7 Read 7 bytes from the device. (14 byte for 16-bit IO)

BRD **Burst read mode**. Setting this field to be logic-1 enables the data read operation. The NFI core will issue read cycles to retrieve data from the device when the data FIFO is not full or the device is not in the busy state. The NFI core supports consecutive page reading.

BWR **Burst write mode**. Setting to be logic-1 enables the data burst write operation.

SEC_NUM The field represents the sector number to be retrieved from the device or AHB. The valid number ranges from 1 to 8.

NFI+000Ch NAND Flash Access Timing Control register NFI_ACCCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**LCD2NAND**				**PRECS**						**C2R**					
Type	R/W				R/W						R/W					
Reset	F				0F						3F					
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**W2R**				**WH**				**WST**				**RLT**			
Type	R/W				R/W				R/W				R/W			
Reset	F				F				F				F			

This is the timing access control register for the NAND FLASH interface. In order to accommodate operations for different system clock frequency ranges from 13MHz to 61.44MHz, wait states and setup/hold time margin can be configured in this register.

C2R The field represents the minimum required time from NCEB low to NREB low (in step of 2T).

W2R The field represents the minimum required time from NWEB high to NREB low. It's in unit of 2T. So the actual time ranges from 2T to 8T in step of 2T.

WH Write-enable hold-time.
The field specifies the hold time of NALE, NCLE, NCEB signals relative to the rising edge of NWEB. This field is associated with **WST** to expand the write cycle time, and is associated with **RLT** to expand the read cycle time.

RLT Read Latency Time
The field specifies how many wait states to be inserted to meet the requirement of the read access time for the device.
00 No wait state.
01 1T wait state.
10 2T wait state.
11 3T wait state.

WST Write Wait State
The field specifies the wait states to be inserted to meet the requirement of the pulse width of the NWEB signal.
00 No wait state.
01 1T wait state.
10 2T wait state.

11 3T wait state.

PRECS The field represents the minimum required time for CS pre-pulling down before any access to device (in step of 8T).

LCD2NAND Arbitration Wait State

The field specifies the wait states to be inserted for the APB arbitrator when bus user changes.

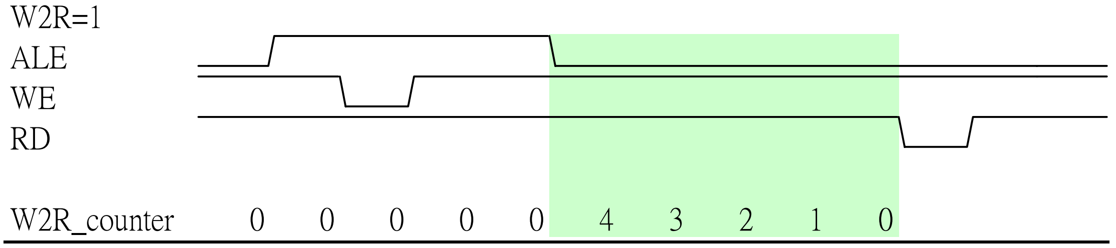
NOTES1 **:** **C2R → 2*C2R + 1**

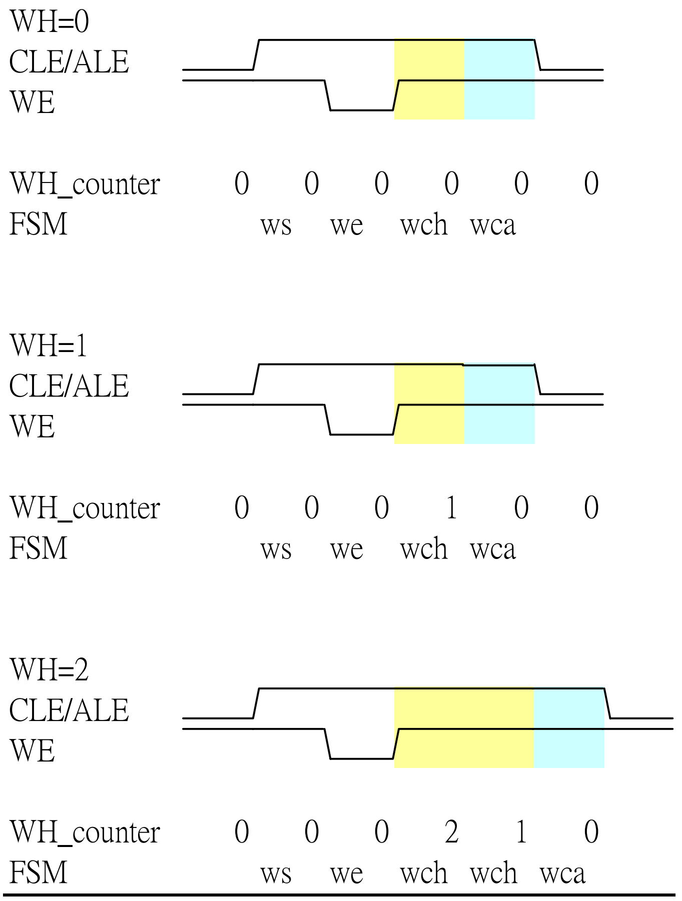
W2R → 2*W2R + 3

WH/RLT/WST → WH/RLT/WST + 1

PRECS → 8*PRECS + 1 **when PRECE>0**

PRECS → 0 **when PRECE=0**





 © 2009 MediaTek Inc.

***NOTES2** : The nli_arbiter behavior need be taken care. For example, in MT6516 the clock of nli_arbiter is 104MHz, and it will sample all nfi signal, such that the RLT must larger than 0.*

***NOTES3** :* **RLT-1(for nli_arbiter with 104MHz) > max(tREA, tRP) + {nre to PAD_NRE delay} + max({PAD_IO to ndi delay}) + {PAD_NRE to NAND input delay} + max({NAND data outpu to PAD_IO delay})**

take MT6516 & K9F2G08U0A 3.3V for example: max(tREA, tRP) = 20ns

{nre to PAD_NRE delay} = 7ns

max({PAD_IO to ndi delay}) = 3.4ns

NFI +0010h NFI Interrupt Enable Register NFI_INTR_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										AHB_DONE_EN	ACCESS_LOCK_EN	BUSY_RETURN_EN	ERASE_DONE_EN	RESET_DONE_EN	WR_DONE_EN	RD_DONE_EN
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

This register controls the activity for the interrupt sources. These enable should be turned on only while SW expects the corresponding interrupt will occur.

RD_DONE_EN The single page read completion interrupt enable.

WR_DONE_EN The single page write completion interrupt enable.

RESET_DONE_EN The reset completion interrupt enable.

ERASE_DONE_EN The erase completion interrupt enable.

BUSY_RETURN_EN The busy return interrupt enable.

ACCESS_LOCK_EN The access lock interrupt enable.

AHB_DONE_EN The done interrupt enable for AHB mode.

NFI +0014h NFI Interrupt Status Register NFI_INTR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										AHB_DONE	ACCESS_LOCK	BUSY_RETURN	ERASE_DONE	RESET_DONE	WR_DONE	RD_DONE
Type										RC	RC	RC	RC	RC	RC	RC
Reset										0	0	0	0	0	0	0

The register indicates the status of all the interrupt sources. Read this register will clear all interrupts.

RD_COMPLETE Indicates that the single page read operation is completed.

WR_COMPLETE Indicates that the write operation is completed.

RESET_COMPLETE Indicates that the reset operation is completed.

ERASE_COMPLETE Indicates that the erase operation is completed.

BUSY_RETURN Indicates that the device state returns from busy by inspecting the R/B# pin.

ACCESS_LOCK Indicates that the operation is invalid and the address range is locked.

AHB_DONE Indicates that the AHB operation is completed.

***NOTES** :* **Access_lock & AHB_done will be reset when issue nfi_reset.**

NFI +0020h NFI Command register NFI_CMD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CMD							
Type									R/W							
Reset									0							

This is the command input register. The user should write this register to issue a command. Please refer to device datasheet for the command set. Before write the command, please check out the settings for register **NFI_CON**.

CMD Command word.

NFI +0030h NFI Address Length Register NFI_ADDRNOB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										ROW_ADDR_NOB				COL_ADDR_NOB		
Type										R/W				R/W		
Reset										0				0		

This register represents the number of bytes corresponding to current command. The each valid number of bytes ranges from 0 to 4. The address format depends on what device to be used and what commands to be applied. The NFI core is made transparent to those different situations except that the user has to define the number of bytes.

The user should write the target address to the address register **NFI_COLADDR** and **NFI_ROWADDR** before programming this register.

COL_ADDR_NOB Number of bytes for the column address

ROW_ADDR_NOB Number of bytes for the row address

NOTES **: When Lock_en is turn on, these number will be automatic set to pre-defined value as command is 0x6X or 0x8X.**

NFI +0034h NFI Column Address Register NFI_COLADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COL_ADDR3								COL_ADDR2							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COL_ADDR1								COL_ADDR0							
Type	R/W								R/W							
Reset	0								0							

This defines the 4 bytes of the column address field to be applied to the device. Since the device bus width is 1 byte, the NFI core arranges the order of address data to be least significant byte first. The user should put the first address byte in the field **ADDR0**, the second byte in the field **ADDR1**, and so on.

COL_ADDRn The n-th column address byte.

NFI +0038h NFI Row Address Register NFI_ROWADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ROW_ADDR3								ROW _ADDR2							
Type	R/W								R/W							

Reset	0								0							
Bit	15	14	15	14	15	14	15	14	15	14	15	14	15	14	15	14
Name	**ROW _ADDR1**								**ROW _ADDR0**							
Type	R/W								R/W							
Reset	0								0							

This defines the 4 bytes of the row address field to be applied to the device. Since the device bus width is 1 byte, the NFI core arranges the order of address data to be least significant byte first. The user should put the first address byte in the field **ADDR0**, the second byte in the field **ADDR1**, and so on.

ROW_ADDRn The n-th row address byte.

NFI +0040h NFI Data Transfer Start Trigger Register NFI_STRDATA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR_ DATA**
Type																WO
Reset																0

This register controls the activity for the interrupt sources.

STR_DATA This signal triggers the data transfer for read or write. It only takes effect as custom operation mode

NFI +0050h NFI Write Data Buffer NFI_DATAW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DW3**								**DW2**							
Type	WO								WO							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DW1**								**DW0**							
Type	WO								WO							
Reset	0								0							

This is the write port of the data FIFO. It supports word access. The least significant byte **DW0** is to be programmed to the device first, then **DW1**, and so on.

DW3 Write data byte 3.

DW2 Write data byte 2.

DW1 Write data byte 1.

DW0 Write data byte 0.

NFI +0054h NFI Read Data Buffer NFI_DATAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DR3**								**DR2**							
Type	RO								RO							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DR1**								**DR0**							
Type	RO								RO							
Reset	0								0							

This is the read port of the data FIFO. It supports word access. The least significant byte **DR0** is the first byte read from the device, then **DR1**, and so on.

DR3 Read data byte 3.

DR2 Read data byte 2.

DR1 Read data byte 1.

DR0 Read data byte 0.

NFI +0060h NFI Status NFI_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				NAND_FSM									NFI_FSM			
Type				RO									RO			
Reset				0									0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				READ_EMPTY			BUSY2READY	BUSY				ACCESS_LOCK	DATAW	DATAR	ADDR	CMD
Type				RO			RO	RO				RO	RO	RO	RO	RO
Reset				1			0	0				0	0	0	0	0

This register represents the NFI core control status including command mode, address mode, data program and read mode. The user should poll this register for the end of those operations.

*The value of **BUSY/NAND_BUSY** bit depends on the GPIO configuration. If GPIO is configured for NAND flash application, the reset value should be 0, which represents that NAND flash is in idle status. When the NAND flash is busy, the value will be 1.

BUSY Synchronized busy signal from the NAND flash. It's read-only. This signal is sampled from NFI

BUSY2READY It's read-only. This signal indicates NAND from busy to ready state and it will be reset after nfi_reset or write command/address.

Notes: device busy duration must larger than 2/nfi_bclk_ck

DATAW The NFI core is in data write mode.

DATAR The NFI core is in data read mode.

ADDR The NFI core is in address mode.

CMD The NFI core is in command mode.

ACCESS_LOCK The access range is locked for erase or program .

READ_EMPTY Empty page indication during read operation, **include all data, FDM and parity for all sectors**

NFI_FSM The field represents the state of NFI internal FSM.

- **0000** idle.
- **0001** reset. Reset command to ready
- **0010** read busy.
- **0011** read data.
- **0100** program busy
- **0101** program data. Input data command to program command
- **1000** erase busy. Erase command to ready
- **1001** erase data. Erase command 1 to erase command 2

1111 custom mode
1110 custom mode for data access
Others Reserved

NAND_FSM The field represents the state of NAND interface FSM.
00000 IDLE. idle.
11100 PRE_CS. Pre CS state.
00101 CMD_WRST. command write set up
00110 CMD_WR. Command write enable.
00111 CMD_WRHD. Command write hold.
00100 CMD_WRRDY
01001 ADDR_WRST. Address write set up
01010 ADDR_WR. Address write enable
01011 ADDR_WRHD. Address write hold
01000 ADDR_WRRDY.
01100 CA2DEXT. Command address write extension.
10001 DATA_RDST. Data read set up.
10010 DATA_RD. Data read enable.
10011 DATA_RDHD. Data read hold.
11000 DATA_WRRDY.
11001 DATA_WRST. Data write set up.
11010 DATA_WR. Data write enable.
11011 DATA_WRHD. Data write hold.
Others Reserved

NFI +0064h NFI FIFO Status NFI_FIFOSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WR_FULL	WR_EMPTY		WR_REMAIN					RD_FULL	RD_EMPTY		RD_REMAIN				
Type	RO	RO		RO					RO	RO		RO				
Reset	0	1		0					0	1		0				

The register represents the status of the data FIFO. The FIFO top and bottom pointer of read & write will be reset when issue “command” to NAND Flash

WR_FULL Data FIFO full in burst write mode.
WR_EMPTY Data FIFO empty in burst write mode.
RD_FULL Data FIFO full in burst read mode.
RD_EMPTY Data FIFO empty in burst read mode.
RD_REMAIN Data FIFO remaining byte number in burst read mode.
WR_REMAIN Data FIFO remaining byte number in burst write mode.

NFI +0068h NFI Lock Status NFI_LOCKSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	ACCESS_LOCK15	ACCESS_LOCK14	ACCESS_LOCK13	ACCESS_LOCK12	ACCESS_LOCK11	ACCESS_LOCK10	ACCESS_LOCK09	ACCESS_LOCK08	ACCESS_LOCK07	ACCESS_LOCK06	ACCESS_LOCK05	ACCESS_LOCK04	ACCESS_LOCK03	ACCESS_LOCK02	ACCESS_LOCK01	ACCESS_LOCK00
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register represents the lock status for each lock range.

If any access_lockxx happens, the nfi core will automatic issue a reset (0xFF) command to NAND device.

ACCESS_LOCKn The access command violates the locking range n

NFI+0070h NFI Page Address Counter Register NFI_ADDRCNTR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEC_CNTR						SEC_ADDR									
Type	RO						RO									
Reset	0						0									

The register represents the current read/write address with respect to initial address input. It counts in unit of byte. In page read and page program operation, the address should be the same as that in the state machine in the target device.

SEC_ADDR The address count of 512 main data and spare data for each sector.

SEC_CNTR The sector count.

NFI+0080h NFI AHB Start Address Register NFI_STRADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STR_ADDR															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STR_ADDR															
Type	R/W															
Reset	0															

The register represents the start address for AHB to access EMI. These memory from the start address is used to put read data from NAND or write data to NAND in AHB mode

STR_ADDR The start address of EMI for both read or write in AHB mode. **This address must be 4-byte aligned.**

NOTES : **If start address of any sector data is not 4-byte aligned (especially disable AUTO_FMT_EN and spare size is not 16 bytes), it must be 1-byte access for AHB mode (set BYTE_RW for 1-byte access) whether NFIECC or Auto-correction is used or not.**

NFI+0084h NFI AHB Byte Length Register NFI_BYTELEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				BYTE_LEN												
Type				RO												
Reset				0												

The register represents the current length for AHB to access EMI.

BYTE_LEN The current length of EMI for both read or write in AHB mode. (byte unit)

NOTES : **This register is used for polling AHB activity**

NFI+0090h NFI device select register NFI_CSEL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**CSEL**
Type																R/W
Reset																0

The register is used to select the target device. It decides which CEB pin to be functional. This is useful while using the high-density device.

CSEL Chip select. The value defaults to 0.

0 Device 1 is selected.

1 Device 2 is selected.

NOTES : **This register is latched as issue CMD**

NFI+0094h NFI IO Control register NFI_IOCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**NLD_PD**
Type																R/W
Reset																0

Data bus pull down when no use.

NLD_PD data bus pull down when no use.

0 disable.

1 enable.

NFI +00A0h NFI Least FDM Data for Sector 0 Register NFI_FDM0L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FDM0_3**								**FDM0_2**							
Type	R/W								R/W							
Reset	ff								ff							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FDM0_1**								**FDM0_0**							
Type	R/W								R/W							
Reset	ff								ff							

This register represents the Least FDM data for the sector 0. Since the device bus width is 1 byte, the NFI core arranges the order of address data to be least significant byte first. The user should put the first address byte in the field **FDM0_0**, the second byte in the field **FDM0_1**, and so on. It will be reset to 0xFF when issue **NFI_Reset**.

FDM0_n The n-th FDM byte data for sector 0.

NFI +00A4h NFI Most FDM Data for Sector 0 Register NFI_FDM0M

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FDM0_7**								**FDM0_6**							
Type	R/W								R/W							

Reset	ff								ff							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FDM0_5**								**FDM0_4**							
Type	R/W								R/W							
Reset	ff								ff							

This register represents the Most FDM data for the sector 0. Since the device bus width is 1 byte, the NFI core arranges the order of address data to be least significant byte first. The user should put the first address byte in the field **FDM0_4**, the second byte in the field **FDM0_7**, and so on. It will be reset to 0xFF when issue **NFI_Reset**.

FDM0_n The n-th FDM byte data for sector 0.

Register Address	Register Function	Acronym
NFI +00A8h	NFI Least FDM Data for Sector 1	NFI_FDM1L
NFI +00ACh	NFI Most FDM Data for Sector 1	NFI_FDM1M
NFI +00B0h	NFI Least FDM Data for Sector 2	NFI_FDM2L
NFI +00B4h	NFI Most FDM Data for Sector 2	NFI_FDM2M
NFI +00B8h	NFI Least FDM Data for Sector 3	NFI_FDM3L
NFI +00BCh	NFI Most FDM Data for Sector 3	NFI_FDM3M
NFI +00C0h	NFI Least FDM Data for Sector 4	NFI_FDM4L
NFI +00C4h	NFI Most FDM Data for Sector 4	NFI_FDM4M
NFI +00C8h	NFI Least FDM Data for Sector 5	NFI_FDM5L
NFI +00CCh	NFI Most FDM Data for Sector 5	NFI_FDM5M
NFI +00D0h	NFI Least FDM Data for Sector 6	NFI_FDM6L
NFI +00D4h	NFI Most FDM Data for Sector 6	NFI_FDM6M
NFI +00D8h	NFI Least FDM Data for Sector 7	NFI_FDM7L
NFI +00DCh	NFI Most FDM Data for Sector 7	NFI_FDM7M

Table 42 ***NFI FDM Data Register Table***

NFI +0100h NFI Lock Enable Register NFI_LOCK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**LOCK_ON**
Type																R/W1
Reset																0

This register enable the lock function of NFI .

These setting can only be set once after reset chip.

LOCK_ON Enable the lock checking process for any lock set.

- **0** Disable lock checking process.
- **1** Enable lock checking process.

NFI +0104h NFI Lock Control Register NFI_LOCK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LOCK15_CS	LOCK15_EN	LOCK14_CS	LOCK14_EN	LOCK13_CS	LOCK13_EN	LOCK12_CS	LOCK12_EN	LOCK11_CS	LOCK11_EN	LOCK10_CS	LOCK10_EN	LOCK09_CS	LOCK09_EN	LOCK08_CS	LOCK08_EN
Type	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LOCK07_CS	LOCK07_EN	LOCK06_CS	LOCK06_EN	LOCK05_CS	LOCK05_EN	LOCK04_CS	LOCK04_EN	LOCK03_CS	LOCK03_EN	LOCK02_CS	LOCK02_EN	LOCK01_CS	LOCK01_EN	LOCK00_CS	LOCK00_EN
Type	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register control the lock function of NFI .

These setting can only be set once after reset chip.

LOCKn_EN Enable the lock checking process of lock set n. Before it takes effect, the **LOCK_ON** must be turned on.

- **0** Disable Lock Range check for set n.
- **1** Enable Lock Range check for set n.

LOCKn_CS Indicate the lock checking process of lock set.n for CS0 or CS1

- **0** Lock range check of set n for CS0.
- **1** Lock range check of set n for CS1.

NFI +0108h NFI Address Format for Lock Register NFI_LOCKANOB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PROG_RADD_NOB				PROG_CADD_NOB				ERASE_RADD_NOB				ERASE_CADD_NOB		
Type		R/W1				R/W1				R/W1				R/W1		
Reset		0				0				0				0		

This register represents the number of bytes corresponding to erase and program command. The each valid number of bytes ranges from 0 to 4. The address format depends on what device to be used and what commands to be applied. The NFI core will force these setting during some command operation(8X or 6X).

These setting can only be set once after reset chip.

ERASE_CADD_NOB Number of bytes for the column address for erase operation (command is 8'h6X)

ERASE_RADD_NOB Number of bytes for the row address for erase operation (command is 8'h6X)

PROG_CADD_NOB Number of bytes for the column address for program operation (command is 8'h8X)

PROG_RADD_NOB Number of bytes for the row address for program operation (command is 8'h8X)

NFI +0110h NFI Row Start Address for Lock Set00 Register NFI_LOCK00ADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LOCK00_ROW3								LOCK00_ROW2							
Type	R/W1								R/W1							
Reset	0								0							
Bit	15	14	15	14	15	14	15	14	15	14	15	14	15	14	15	14
Name	LOCK00_ROW1								LOCK00_ROW0							
Type	R/W1								R/W1							

Reset	0	0

This defines the 4 bytes of the row start address field to be locked range for the device.

These setting can only be set once after reset chip.

LOCK00_ROWn The n-th row start address byte to be locked for lock set 0.

NFI +0114h NFI Row Address Format for Lock Set00 Register NFI_LOCK00FMT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LOCK00_FMT3								LOCK00_ FMT 2							
Type	R/W1								R/W1							
Reset	0								0							
Bit	15	14	15	14	15	14	15	14	15	14	15	14	15	14	15	14
Name	LOCK00_ FMT 1								LOCK00_ FMT 0							
Type	R/W1								R/W1							
Reset	0								0							

This defines the 4 bytes format of the row address field to be locked range for the device.

These setting can only be set once after reset chip.

The MSB unused range must be set to 0 for LOCKxxFMT.

LOCK00_FMTn The n-th row address format byte to be locked for lock set 0.

NOTES : **The Real Lock Range → Lock0_Rown && Lock0_FMTn**

Register Address	Register Function	Acronym
NFI +0118h	NFI Row Start Address for Lock Set 01	NFI_LOCK01ADD
NFI +011Ch	NFI Row Address Format for Lock Set 01	NFI_LOCK01FMT
NFI +0120h	NFI Row Start Address for Lock Set 02	NFI_LOCK02ADD
NFI +0124h	NFI Row Address Format for Lock Set 02	NFI_LOCK02FMT
NFI +0128h	NFI Row Start Address for Lock Set 03	NFI_LOCK03ADD
NFI +012Ch	NFI Row Address Format for Lock Set 03	NFI_LOCK03FMT
NFI +0130h	NFI Row Start Address for Lock Set 04	NFI_LOCK04ADD
NFI +0134h	NFI Row Address Format for Lock Set 04	NFI_LOCK04FMT
NFI +0138h	NFI Row Start Address for Lock Set 05	NFI_LOCK05ADD
NFI +013Ch	NFI Row Address Format for Lock Set 05	NFI_LOCK05FMT
NFI +0140h	NFI Row Start Address for Lock Set 06	NFI_LOCK06ADD
NFI +0144h	NFI Row Address Format for Lock Set 06	NFI_LOCK06FMT
NFI +0148h	NFI Row Start Address for Lock Set 07	NFI_LOCK07ADD
NFI +014Ch	NFI Row Address Format for Lock Set 07	NFI_LOCK07FMT
NFI +0150h	NFI Row Start Address for Lock Set 08	NFI_LOCK08ADD
NFI +0154h	NFI Row Address Format for Lock Set 08	NFI_LOCK08FMT
NFI +0158h	NFI Row Start Address for Lock Set 09	NFI_LOCK09ADD
NFI +015Ch	NFI Row Address Format for Lock Set 09	NFI_LOCK09FMT
NFI +0160h	NFI Row Start Address for Lock Set 10	NFI_LOCK10ADD

NFI +0164h	NFI Row Address Format for Lock Set 10	NFI_LOCK10FMT
NFI +0168h	NFI Row Start Address for Lock Set 11	NFI_LOCK11ADD
NFI +016Ch	NFI Row Address Format for Lock Set 11	NFI_LOCK11FMT
NFI +0170h	NFI Row Start Address for Lock Set 12	NFI_LOCK12ADD
NFI +0174h	NFI Row Address Format for Lock Set 12	NFI_LOCK12FMT
NFI +0178h	NFI Row Start Address for Lock Set 13	NFI_LOCK13ADD
NFI +017Ch	NFI Row Address Format for Lock Set 13	NFI_LOCK13FMT
NFI +0180h	NFI Row Start Address for Lock Set 14	NFI_LOCK14ADD
NFI +0184h	NFI Row Address Format for Lock Set 14	NFI_LOCK14FMT
NFI +0188h	NFI Row Start Address for Lock Set 15	NFI_LOCK15ADD
NFI +018Ch	NFI Row Address Format for Lock Set 15	NFI_LOCK15FMT

Table 43 ***NFI Lock Range Set Register Table***

NFI +0190h NFI FIFO Content Data 0 NFI_FIFODATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO_DATA0**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO_DATA0**															
Type	RO															
Reset	0															

This register represents the content data 0 of fifo.

NFI +0194h NFI FIFO Content Data 1 NFI_FIFODATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO_DATA1**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO_DATA1**															
Type	RO															
Reset	0															

This register represents the content data 1 of fifo.

NFI +0198h NFI FIFO Content Data 2 NFI_FIFODATA2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO_DATA2**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO_DATA2**															
Type	RO															
Reset	0															

This register represents the content data 2 of fifo.

NFI +019Ch NFI FIFO Content Data 3 NFI_FIFODATA3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFO_DATA3															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO_DATA3															
Type	RO															
Reset	0															

This register represents the content data 3 of fifo.

2.24.3 Device Programming Sequence

This section lists the program sequences for the NAND flash operations.

2.24.3.1 Block Erase

*NFI_CON = 0x3; // reset nfi core
*NFI_INTR_EN = 0x8; // enable erase complete interrupt
*NFI_CNFG = 0x4000; // erase operation
*NFI_CMD = 0x60; // erase first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CMD = 0xD0; // erase second cycle command

// Then, wait for the erase complete interrupt.

2.24.3.2 Page Program

*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_FDMxx = all FDM data; // FDM data content
*NFI_INTR_EN = 0x2; // enable write complete interrupt
*NFI_CNFG = 0x3340; // program operation
*NFI_CMD = 0x80; // page program first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CON = 0x8200; // Set burst write mode

// After MCU / AHB writing a page of bytes,

NFI_CMD = 0x10; // page program second cycle command

// Then, wait for the page program complete interrupt.

2.24.3.3 Page Read

```
*NFI_CON = 0x3;                         // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_INTR_EN = 0x1;                     // enable read complete
*NFI_CNFG = 0x1342;                     // read operation
*NFI_CMD = 0x0;                         // page read command
*NFI_COLADDR = address;                 // address
*NFI_ROWADDR = address;                 // address
*NFI_ADDRNOB = cycle number;// number of page address
while(*NFI_STA == 0);          // wait for the address to be programmed
*NFI_CMD = 0x30;                        // page read second cycle command
*NFI_CON = 0x8100;                      // Set burst write mode
```

// Then, use MCU / AHB to read a page of bytes,

2.24.3.4 Read ID / Read Status

```
*NFI_CMD = 0x70;                        // read status command
*NFI_CON = cycle number;       // Set single word read for n byte
while(*NFI_STA == 0);          // wait for the command to be programmed
status = *NFI_DATAR;           // read the single byte of status
```

2.24.3.5 Reset

```
*NFI_INTR_EN = 0x4;                     // enable reset complete
*NFI_CMD = 0xff;                        // reset command
```

// Then, wait for the reset complete interrupt.

2.24.3.6 Multi Block Erase

```
*NFI_CON = 0x3;                         // reset nfi core
*NFI_INTR_EN = 0x10;           // enable ready busy interrupt
*NFI_CNFG = 0x6000;                     // custom operation
*NFI_CMD = 0x60;                        // erase first cycle command
*NFI_COLADDR = address;                 // address
*NFI_ROWADDR = address;                 // address
*NFI_ADDRNOB = cycle number;// number of page address
while(*NFI_STA == 0);          // wait for the address to be programmed
*NFI_CMD = 0x60;                        // erase first cycle command
*NFI_COLADDR = address;                 // address
*NFI_ROWADDR = address;                 // address
*NFI_ADDRNOB = cycle number;// number of page address
while(*NFI_STA == 0);          // wait for the address to be programmed
```

NFI_CMD = 0xD0;** **// erase second cycle command

// Then, wait for the ready busy interrupt.

2.24.3.7 Multi Page with Data Cache Program

*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_FDMxx = all FDM data; // FDM data content
*NFI_INTR_EN = 0x10; // enable ready busy interrupt
*NFI_CNFG = 0x6340; // custom operation
*NFI_CMD = 0x80; // page program first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CON = 0x8200;** **// Set burst write mode

NFI_STRDATA = 0x1;** **// start data transfer

// After MCU / AHB writing a page of bytes,

NFI_CMD = 0x11;** **// page program second cycle command

// Then, wait for the ready busy complete interrupt.
*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_FDMxx = all FDM data; // FDM data content
*NFI_INTR_EN = 0x10; // enable ready busy interrupt
*NFI_CNFG = 0x6340; // custom operation
*NFI_CMD = 0x80; // page program first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CON = 0x8200;** **// Set burst write mode

NFI_STRDATA = 0x1;** **// start data transfer

// After MCU / AHB writing a page of bytes,

NFI_CMD = 0x15;** **// page program second cycle command

// Then, wait for the ready busy complete interrupt.
*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_FDMxx = all FDM data; // FDM data content
*NFI_INTR_EN = 0x10; // enable ready busy interrupt
*NFI_CNFG = 0x6340; // custom operation

*NFI_CMD = 0x80; // page program first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CON = 0x8200; // Set burst write mode

NFI_STRDATA = 0x1; // start data transfer

// After MCU / AHB writing a page of bytes,

NFI_CMD = 0x11; // page program second cycle command

// Then, wait for the ready busy complete interrupt.
*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_FDMxx = all FDM data; // FDM data content
*NFI_INTR_EN = 0x10; // enable ready busy interrupt
*NFI_CNFG = 0x6340; // custom operation
*NFI_CMD = 0x80; // page program first cycle command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed

NFI_CON = 0x8200; // Set burst write mode

NFI_STRDATA = 0x1; // start data transfer

// After MCU / AHB writing a page of bytes,

NFI_CMD = 0x10; // page program second cycle command

// Then, wait for the ready busy complete interrupt.

2.24.3.8 Multi Page with Data Cache Read

*NFI_CON = 0x3; // reset nfi core
*NFI_PAGEFMT = page format; // device page format
*NFI_INTR_EN = 0x10; // enable ready busy complete
*NFI_CNFG = 0x6342; // custom operation
*NFI_CMD = 0x0; // page read command
*NFI_COLADDR = address; // address
*NFI_ROWADDR = address; // address

NFI_ADDRNOB = cycle number;// number of page address

while(*NFI_STA == 0); // wait for the address to be programmed
*NFI_CMD = 0x30; // page read second cycle command
// Then, wait for the ready busy complete interrupt.

Intr_status = *NFI_INTR; // read the interrupt status
*NFI_CMD = 0x31; // page read with data cache command
// Then, wait for the ready busy complete interrupt.
Intr_status = *NFI_INTR; // read the interrupt status

NFI_CON = 0x8100; // Set burst write mode

NFI_STRDATA = 0x1; // start data transfer

// Then, use MCU / AHB to read a page of bytes,
*NFI_CON = 0x3; // reset nfi core
*NFI_INTR_EN = 0x10; // enable ready busy complete
*NFI_CNFG = 0x6342; // custom operation
*NFI_CMD = 0x31; // page read command
// Then, wait for the ready busy complete interrupt.

NFI_CON = 0x8100; // Set burst write mode

NFI_STRDATA = 0x1; // start data transfer

// Then, use MCU / AHB to read a page of bytes,
*NFI_CON = 0x3; // reset nfi core
*NFI_INTR_EN = 0x10; // enable ready busy complete
*NFI_CNFG = 0x6342; // custom operation
*NFI_CMD = 0x3f; // page read command
// Then, wait for the ready busy complete interrupt.

NFI_CON = 0x8100; // Set burst write mode

NFI_STRDATA = 0x1; // start data transfer

// Then, use MCU / AHB to read a page of bytes,

2.24.4 Control and Timing

2.24.4.1 Timing Diagram

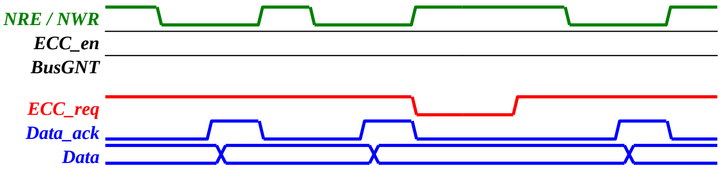


Figure 22 ***Data Interface***

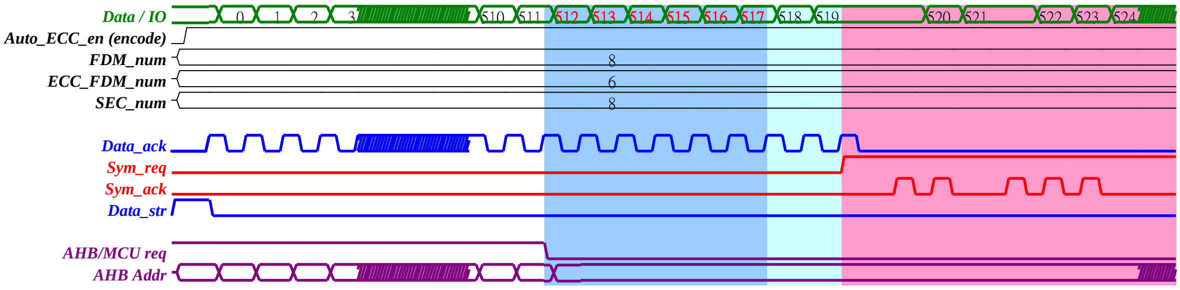


Figure 23 ***Write for Encoding***

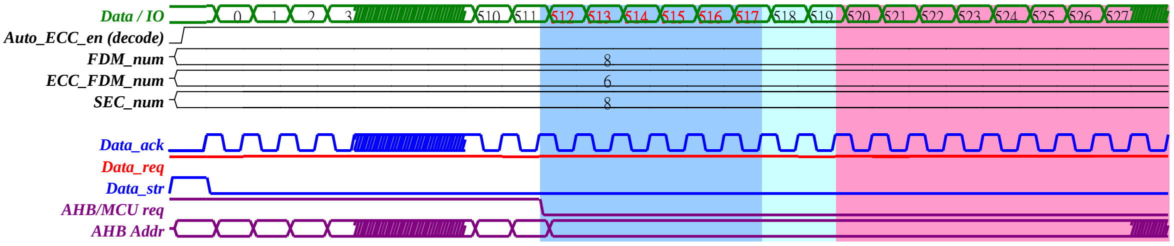


Figure 24 ***Read for Decoding***

2.24.4.2 Finite State Machine Diagram

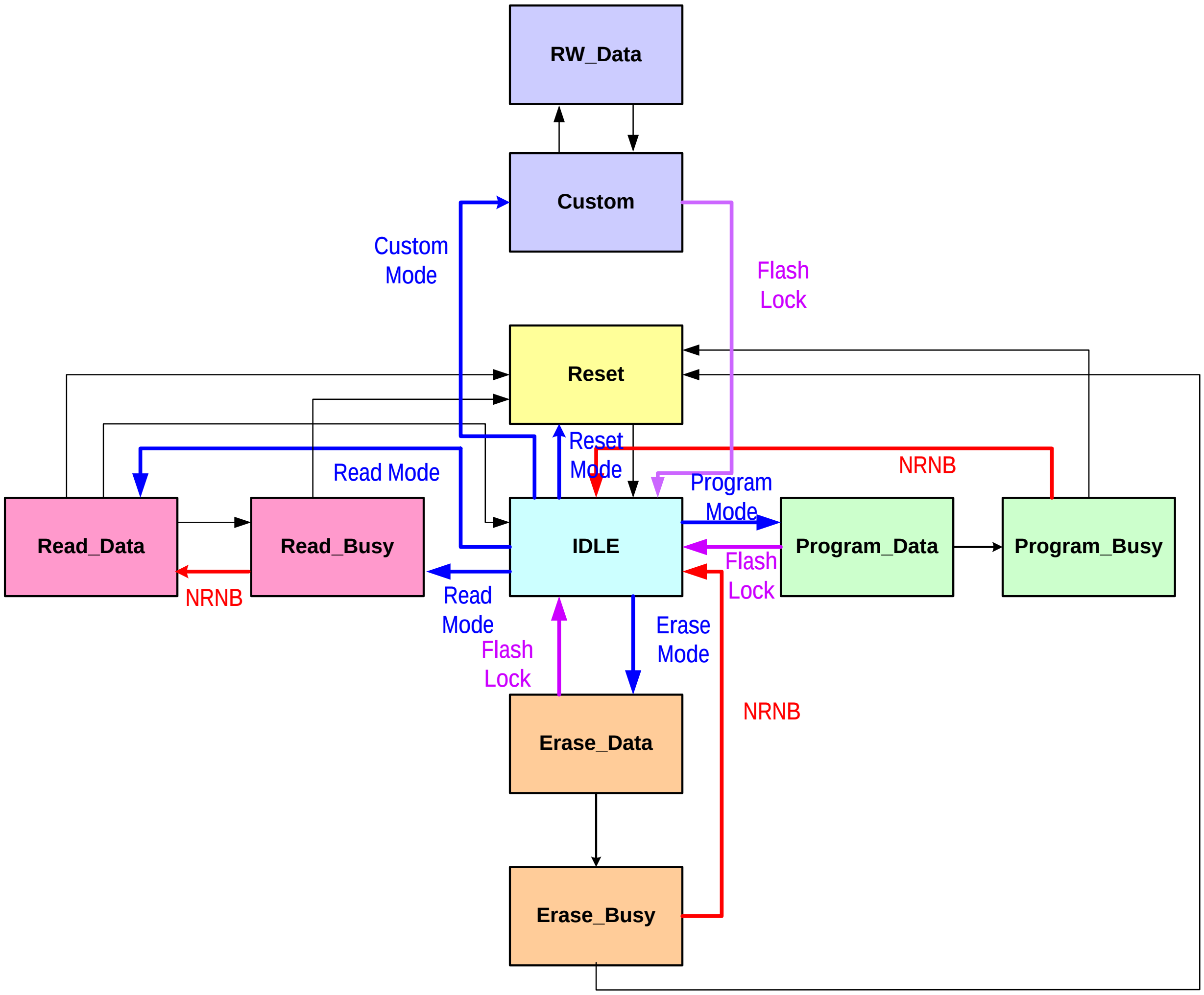


Figure 25 ***NFI Core Control***

2.24.4.3 Device timing control

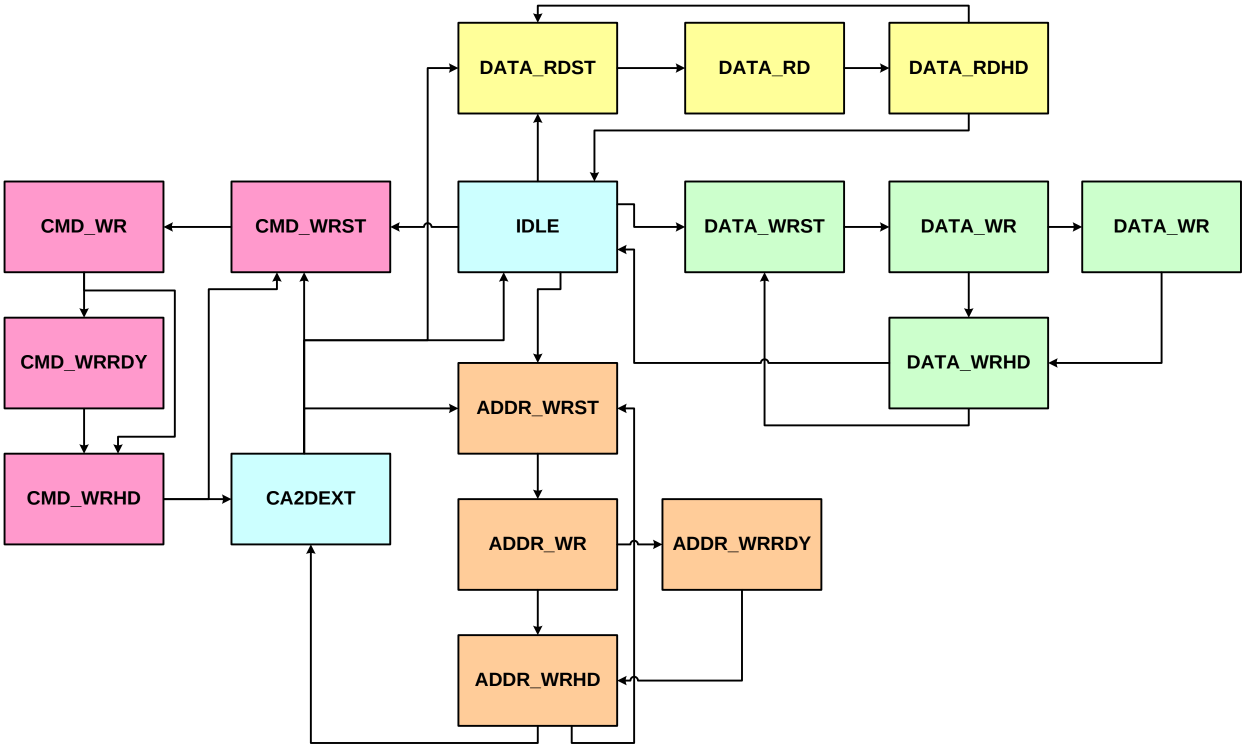


Figure 26 ***NAND Flash Interface Signal Timing Contorl***

This section illustrates the timing diagram.
The ideal timing for write access is listed as listed in **Table 44**.

Parameter	Description	Timing specification	Timing at 13MHz (WST, WH) = (0,0)	Timing at 26MHz (WST, WH) = (0,0)	Timing at 52MHz (WST, WH) = (1,0)
T_{WC1}	***Write cycle time***	**3T + WST + WH**	**230.8ns**	**105.4ns**	**76.9ns**
T_{WC2}	***Write cycle time***	**2T + WST + WH**	**153.9ns**	**76.9ns**	**57.7ns**
T_{DS}	Write data setup time	**1T + WST**	**76.9ns**	**38.5ns**	**38.5ns**
T_{DH}	***Write data hold time***	**1T + WH**	**76.9ns**	**38.5ns**	**19.2ns**
T_{WP}	Write enable time	**1T + WST**	**76.9ns**	**38.5ns**	**38.5ns**
T_{WH}	***Write high time***	**1T + WH**	**76.9ns**	**38.5ns**	**19.2ns**
T_{CLS}	***Command latch enable setup time***	**1T**	**76.9ns**	**38.5ns**	**19.2ns**
T_{CLH}	***Command latch enable hold time***	**1T + WH**	**76.9ns**	**38.5ns**	**19.2ns**
T_{ALS}	***Address latch enable setup time***	**1T**	**76.9ns**	**38.5ns**	**19.2ns**
T_{ALH}	***Address latch enable hold time***	**1T + WH**	**76.9ns**	**38.5ns**	**19.23ns**

© 2009 MediaTek Inc.

F_{WC}	*Write data rate*	1 / T_{WC2}	6.5Mbytes/s	13Mbytes/s	17.3Mbytes/s

Table 44 ***Write access timing***

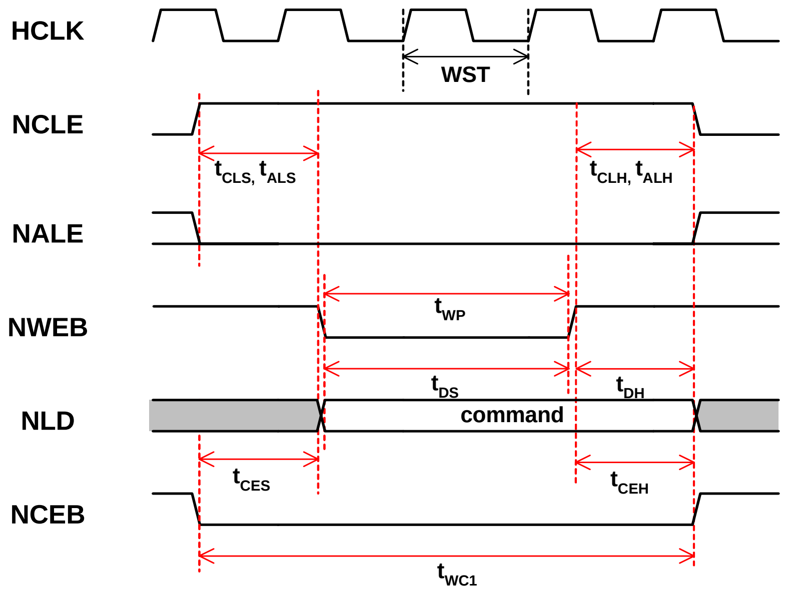


Figure 27 ***Command input cycle (1 wait state).***

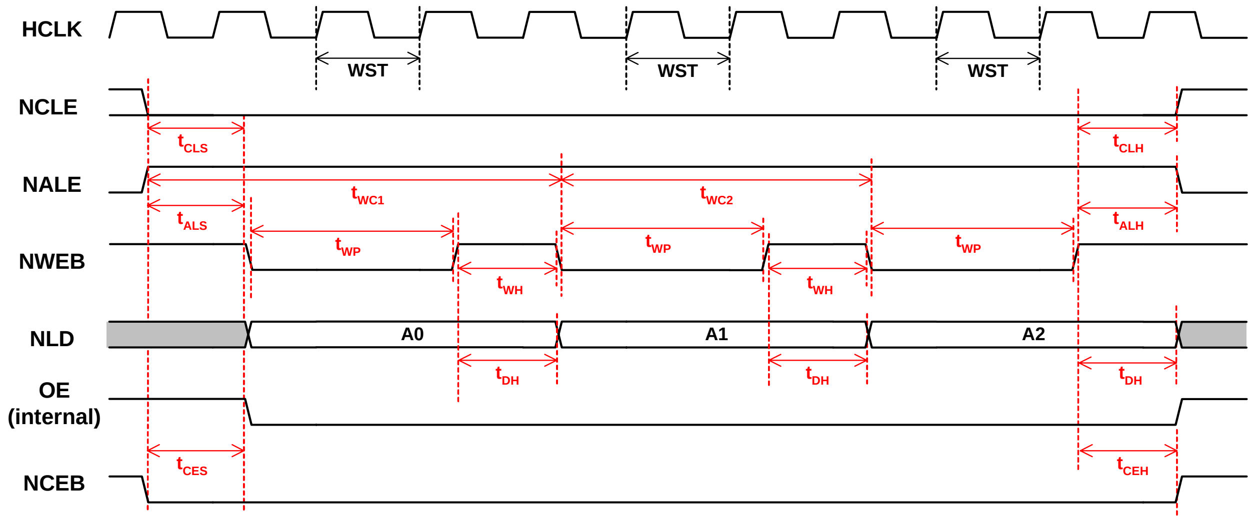


Figure 28 ***Address input cycle (1 wait state)***

© 2009 MediaTek Inc.

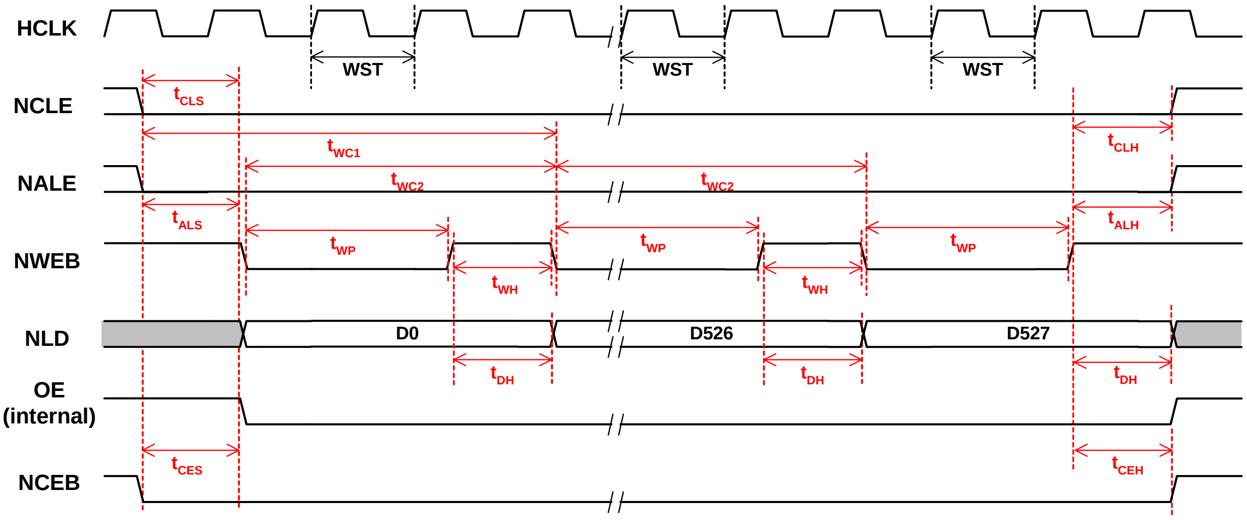


Figure 29 ***Consecutive data write cycles (1 wait state, 0 hold time extension)***

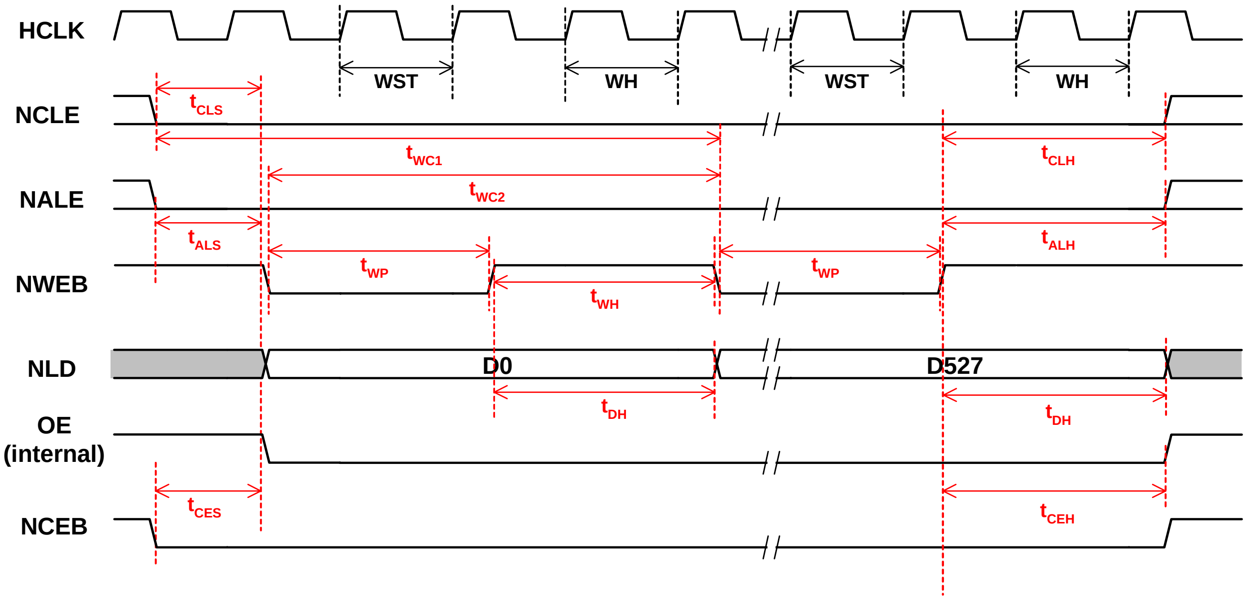


Figure 30 ***Consecutive data write cycles (1 wait state, 1 hold time extension)***

The ideal timing for read access is as listed in **Table 6**.

Parameter	**Description**	**Timing specification**	**Timing at 13MHz (RLT, WH) = (0,0)**	**Timing at 26MHz (RLT, WH) = (1,0)**	**Timing at 52MHz (RLT, WH) = (2,0)**
T_{RC1}	***Read cycle time***	**3T + RLT + WH**	**230.8ns**	**153.8ns**	**96.2ns**
T_{RC2}	***Read cycle time***	**2T + RLT + WH**	**153.9ns**	**115.4ns**	**76.9ns**
T_{DS}	Read data setup time	**1T + RLT**	**76.9ns**	**76.9ns**	**57.7ns**
T_{DH}	***Read data hold time***	**1T + WH**	**76.9ns**	**38.5ns**	**19.2ns**

 © 2009 MediaTek Inc.

T_{RP}	Read enable time	1T + RLT	76.9ns	76.9ns	57.7ns
T_{RH}	*Read high time*	1T + WH	76.9ns	38.5ns	19.2ns
T_{CLS}	*Command latch enable setup time*	1T	76.9ns	38.5ns	19.2ns
T_{CLH}	*Command latch enable hold time*	1T + WH	76.9ns	38.5ns	19.2ns
T_{ALS}	*Address latch enable setup time*	1T	76.9ns	38.5ns	19.2ns
T_{ALH}	*Address latch enable hold time*	1T + WH	76.9ns	38.5ns	19.2ns
F_{RC}	*Write data rate*	1 / T_{RC2}	6.5Mbytes/s	8.7Mbytes/s	13Mbytes/s

Table 45 ***Read access timing***

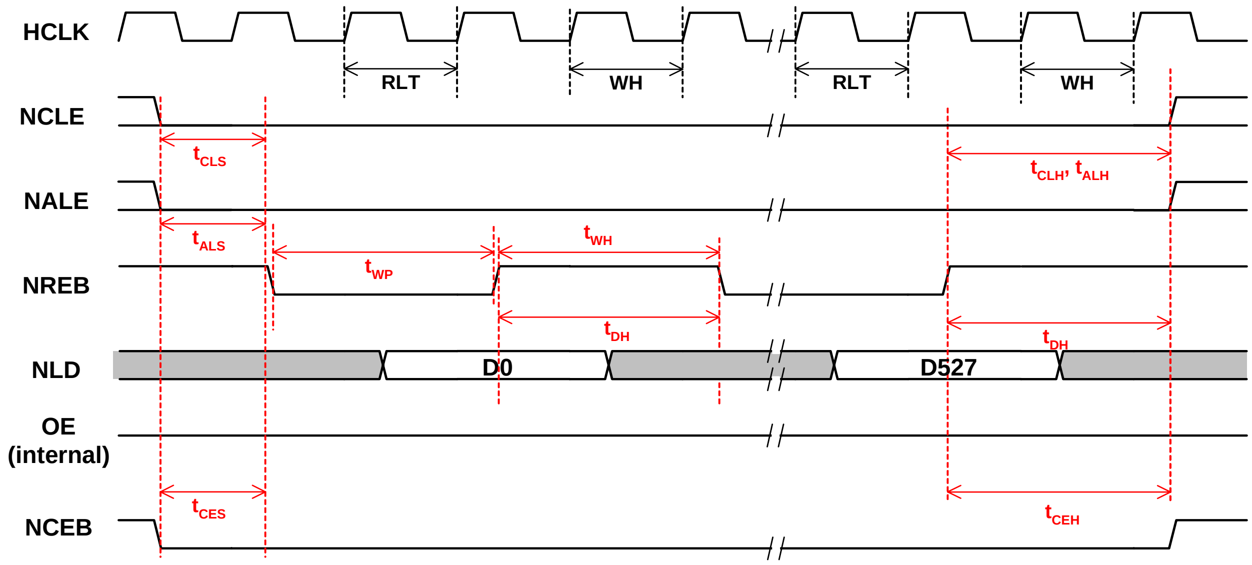


Figure 31 ***Serial read cycle (1 wait state, 1 hold time extension)***

© 2009 MediaTek Inc.

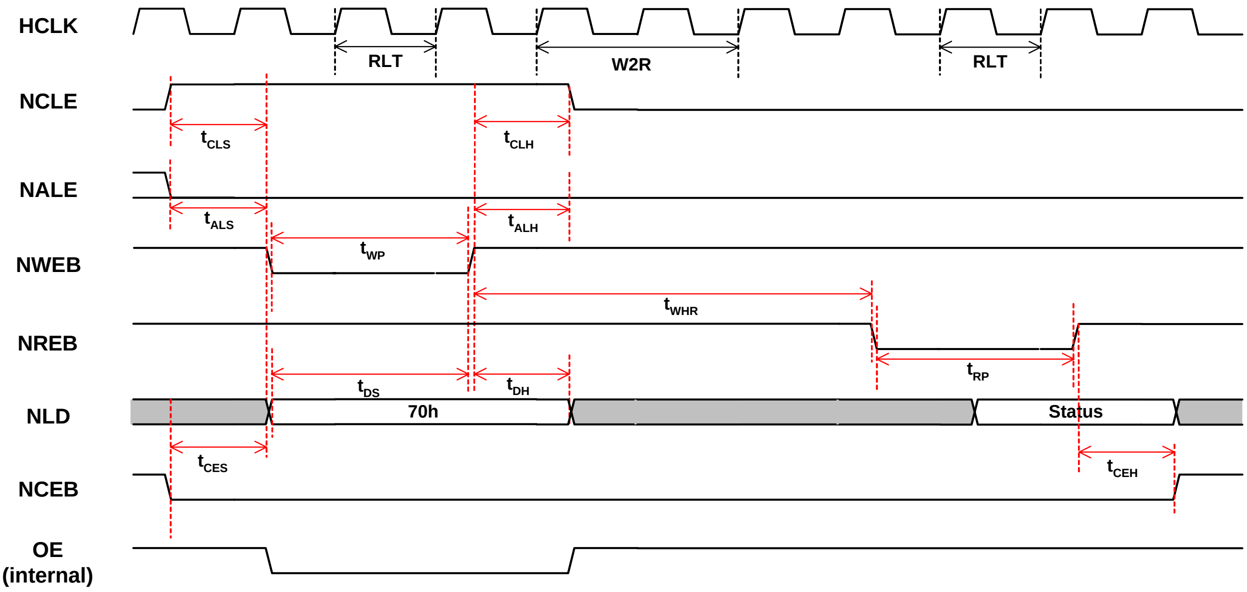


Figure 32 ***Status read cycle (1 wait state)***

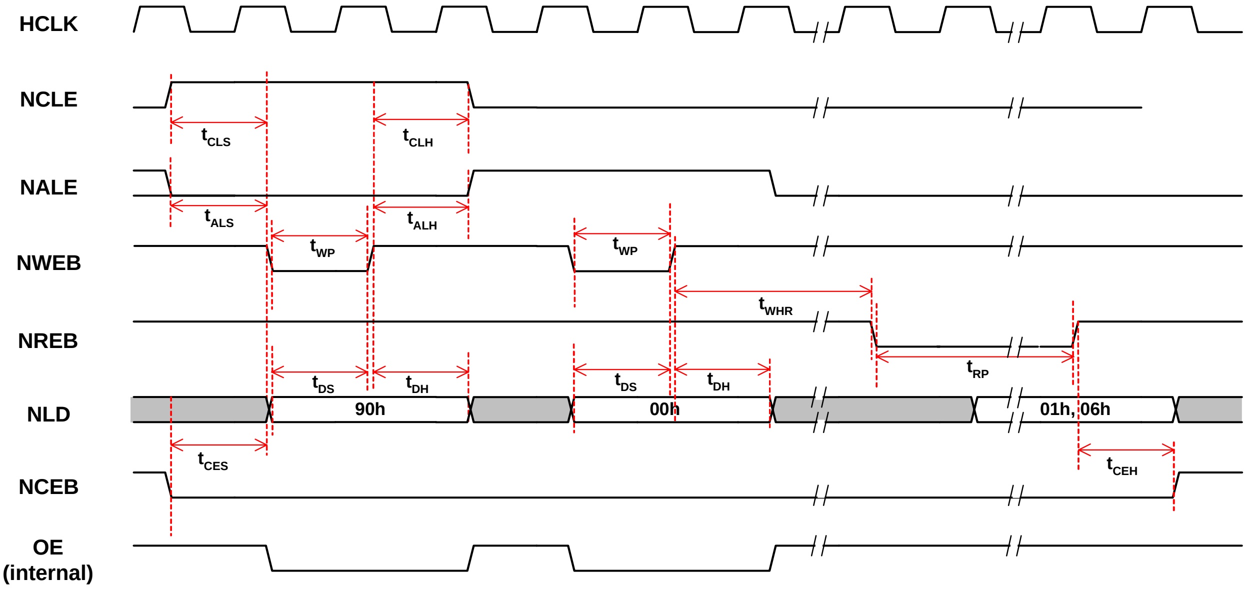


Figure 33 ***ID and manufacturer read (0 wait state)***

© 2009 MediaTek Inc.

2.25 NAND FLASH ECC

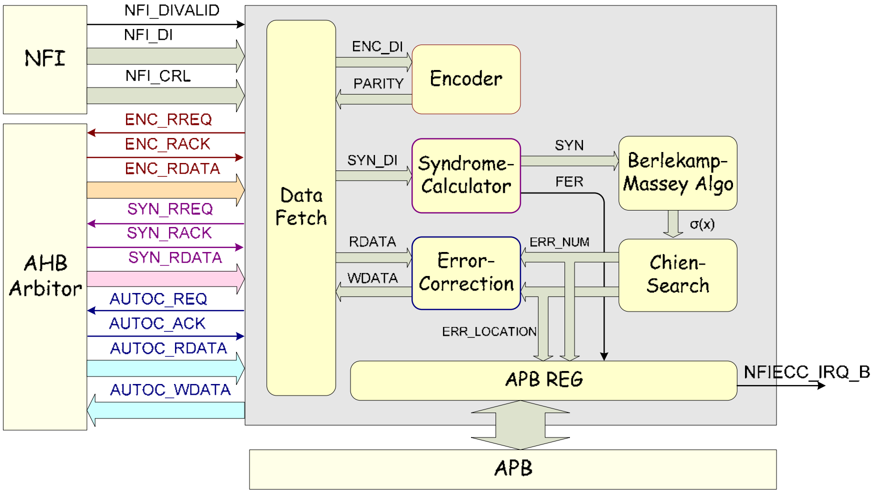


Figure 34 ***Block Diagram of Nand Flash ECC.***

BCH codes are usually referred to as (N,K,t) codes, where N is the number of symbols in a code block, K is the number of data symbols and the t is maximum number of correctable symbol errors in a block. The BCH codec module is implemented in $GF(2^{13})$ defined by primitive polynomial $X^{13}+X^4+X^3+X+1$.

- ECC (BCH code) acceleration is capable of 4/6/8/10/12 bits correction in one full or shorten ECC coded block size which is less than 8192 (<8192bits)
- Support data input in 8/16 bits in NFI mode and 32 bits in AHB mode and works in 104MHz.
- Support encoder and decoder work separately and automatic error correction.

2.25.1 General description

2.25.1.1 Input and Output Interface

2.25.1.1.1 NFI Interface

Signal	Direction	Width	Description
nfi_str	Input	1	Start to encode or decode in NFI mode.
nfi_di_rdy	Output	1	The signal is high when ECC is ready to receive data.
nfi_di	Input	16	Data input port. When nfi_dbyte_en is equal to 0, the valid bits of input data is eight and is filled into the lowest eight bits ([7:0]). The LSB (bit 0) indicates the highest power of input data which will be encoded and decoded first. Otherwise, when nfi_dbyte_en is equal to 1, the valid bits of input data is sixteen bits ([15:0]).
nfi_di_valid	Input	1	The signal is high when input data is valid.
nfi_str_addr	Input	32	The memory address indicates the decoding source data address, used only in the decoding processing of NFI mode.
nfi_fdm_mod	Input	1	The signal indicates the fdm data location.

e			
nfi_sec_num	Input	3	The signal indicates the sector number of input decoding data.
nfi_dbyte_en	Input	1	The signal informs the valid bit of input data, 0 for 8 bits and 1 for16 bits.
nfi_par_req	Input	1	The signal indicates the data has been read and the ECC module will output next data.
nfi_lbyte_en	Input	1	The signal informs the 16 bits of input data, only valid in lower location byte. [7:0]
nfi_mbyte_en	Input	1	The signal informs the 16 bits of output data, only valid in higher location byte. [15:8]
nfi_par_rdy	Output	1	The signal asserts when parity output is ready in nfi_do port.
nfi_parity	Output	16	Encoding data output port. Valid bit-width is same as input bit-width.

Table 1 ***NFI Interface***

2.25.1.1.2 Memory Access Interface

Signal	Direction	Width	Description
enc_rreq	Output	1	Encoder block read request to AHB arbiter.
enc_rack	Input	1	Encoder block read acknowledge signal from AHB arbiter.
enc_raddr	Output	32	Encoder block read address to AHB arbiter.
enc_rdata	Input	32	Encoder block read data from memory.
syn_rreq	Output	1	Syndrome block read request to AHB arbiter.
syn_rack	Input	1	Syndrome block read acknowledge signal from AHB arbiter.
syn_raddr	Output	32	Syndrome block read address to AHB arbiter.
syn_rdata	Input	32	Syndrome block read data from memory.
autoc_req	Output	1	Auto-correction block request to AHB arbiter.
autoc_ack	Input	1	Auto-correction block acknowledge signal from AHB arbiter.
autoc_addr	Output	32	Auto-correction block address to AHB arbiter.
autoc_write	Output	1	Auto-correction block write or read indication to AHB arbiter.
autoc_wdata	Output	32	Auto-correction block write data from decoder.
autoc_rdata	Input	32	Auto-correction block read data from memory.

Table 2 ***Memory Access Interface***

2.25.1.1.3 APB Interface

Signal	Direction	Width	Description
bclk_ck	Input	1	Module clock
pclk_ck	Input	1	APB clock
preset_rstb	Input	1	System reset
penable	Input	1	APB enable
psel	Input	1	APB select of NFIECC

© 2009 MediaTek Inc.

pwrite	Input	1	APB read or write command
paddr [15:0]	Input	16	APB address of NFIECC
pwdata [31:0]	Input	32	APB write data of NFIECC
prdata [31:0]	Output	32	APB read data of NFIECC

Table 3 ***APB Interface***

2.25.1.1.4 Interrupt

Signal	Direction	Width	Description
nfiecc_irq_b	Output	1	Inform encoder or decoder is done. Active low signal

Table 4 ***Interrupt***

2.25.2 Registers Memory Map

2.25.2.1 Registers Memory Map

Software responsibility and controllable functions

Register Address	Acronym	Register Function
NFIECC +0000h	NFIECC_ENCON	Encoder Control
NFIECC +0004h	NFIECC_ENCCNFG	Encoder Configure
NFIECC +0008h	NFIECC _ENCDIADDR	Encoder input data address
NFIECC +000Ch	NFIECC _ENCIDLE	Encoder idle
NFIECC +0010h	NFIECC_ENCPAR0	Encoder Parity output bit.
NFIECC +0014h	NFIECC_ENCPAR1	Encoder Parity output bit.
NFIECC +0018h	NFIECC_ENCPAR2	Encoder Parity output bit.
NFIECC +001Ch	NFIECC_ENCPAR3	Encoder Parity output bit.
NFIECC +0020h	NFIECC_ENCPAR4	Encoder Parity output bit.
NFIECC +0024h	NFIECC_ENCSTA	Encoder Status report
NFIECC +0028h	NFIECC_ENCIRQEN	Encoder IRQ mask.
NFIECC +002Ch	NFIECC_ENCIRQSTA	Encoder IRQ Status report
NFIECC +0100h	NFIECC_DECCON	Decoder Control.
NFIECC +0104h	NFIECC_DECCNFG	Decoding Configuration.
NFIECC +0108h	NFIECC_DECDIADDR	Decoder input data address
NFIECC +010Ch	NFIECC_DECIDLE	Decoder Start Status report
NFIECC +0110h	NFIECC_DECFER	Decoder Stop Status report
NFIECC +0114h	NFIECC_DECENUM	Decoder Stop Status report
NFIECC +0118h	NFIECC_DECDONE	Decoder Stop Status report
NFIECC +011Ch	NFIECC_DECEL0	Decoder Error location 0 report

© 2009 MediaTek Inc.

NFIECC +0120h	NFIECC_DECEL1	Decoder Error location 1 report
NFIECC +0124h	NFIECC_DECEL2	Decoder Error location 2 report
NFIECC +0128h	NFIECC_DECEL3	Decoder Error location 3 report
NFIECC +012Ch	NFIECC_DECEL4	Decoder Error location 4 report
NFIECC +0130h	NFIECC_DECEL5	Decoder Error location 5 report
NFIECC +0134h	NFIECC_DECIRQEN	Decoder IRQ mask.
NFIECC +0138h	NFIECC_DECIRQSTA	Decoder IRQ Status report
NFIECC +013Ch	NFIECC_FDMADDR	first FDM data register address.
NFIECC +0140h	NFIECC_DECFSM	Decoder FSM of all stage and sec_num.
NFIECC +0144h	NFIECC_SYNSTA	Syndrome status.
NFIECC +0148h	NFIECC_NFIDI	NFI input data.
NFIECC +014Ch	NFIECC_SYN0	NFI syndrome data.

Table 5 ***Registers Memory Map Table***

2.25.2.2 Register definition

Timing Definition

- Immediate(Immd) → no buffering of the parameter, update any time when the register is accessed.
- Tsyn → update when syndrome process is finished of an encoding or decoding processor.
- Tel → update when Chien search is finished of a decoding processor.
- Tcorr → update when a error correction processor has done.
- Tdone → update when a decoding processor has done.

NFIECC+0000h NFIECC Encoder Control Register NFIECC_ENCCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ENC_EN**
Type																R/W
Reset																0
Timing																Immd

This register is for Encoder control.

ENC_EN indicates the enable in NFI mode and start to work in AHB mode. In AHB mode, parity bits is remained in the PAR0~PAR4 register field until the ENC_EN is deasserted to 0.

0 means disable the Encode block.

1 means enable the Encode block. In AHB mode, the Encoder starts to fetch data when the register changes **from 0 to 1**. In NFI mode, the register enables the Encode block, and then the Encoder module waits start signal and data from NFI.

NFIECC+0004h NFIECC Configure Register NFIECC_ENCCNFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				ENC_MS												
Type				R/W												
Reset				0												
Timing				Immd												

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ENC_NFI_MODE		ENC_TNUM		
Type												R/W		R/W		
Reset												0		0		
Timing												Immd		Immd		

This register is for NFIECC encoder configuration.

ENC_TNUM indicates the correct capability in one block size.
- **0** means the NFIECC is capable of correct 4 bits in one block size.
- **1** means the NFIECC is capable of correct 6 bits in one block size.
- **2** means the NFIECC is capable of correct 8 bits in one block size.
- **3** means the NFIECC is capable of correct 10 bits in one block size.
- **4** means the NFIECC is capable of correct 12 bits in one block size.

ENC_NFI_MODE indicates the data source from access through AHB bus or from NFI.
- **0** means source data from access through AHB bus.
- **1** means source data from NFI module.

ENC_MS indicates the total bit size of message block **including main data and control(FDM) data in the NFI mode**. The **spare_ECC_num** parameter in old version has been merged into the message block_size parameter. If the block_size is equal to zero, the NFIECC do nothing.

The acceptable **coded block size, which includes data and parity bits size,** is 1~8191bits. Different ENC_TNUM results in different parity bits, and also results in different maximum message block size. The relationship shows in table 6.

T	Maximum (ENC_MS)	Parity bit number
4	8191-52 =8139	4*13=52
6	8191-78 =8113	6*13=78
8	8191-104 =8087	8*13=104
10	8191-130 =8061	10*13=130
12	8191-156 =8035	12*13=156

Table 6 ***Parity bit number and Maximum Message block size Table***

The figure shows the defined block_size. The area of oblique line is the data that need to protect. The parity block is generated by Encoder. The DEC_CS in NFIECC_DECCNFG should include parity bits, thus the ENC_MS and DEC_CS has relationship as : **DEC_CS = ENC_MS + T*13**. T indicates the correct capability.

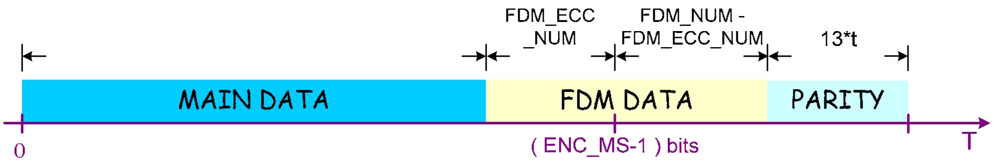


Figure 2 ***The relationship of Block size in NFI mode.***

NFIECC+0008h NFIECC Encoder DI Memory Address Register NFIECC_ENCDIADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ENC_DIADDR**															
Type	R/W															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ENC_DIADDR**															
Type	R/W															
Reset	0															
Timing	Immd															

The register indicates the data start address of input data to the Encoder AHB mode.

ENC_DIADDR indicates the memory address of input data to Encoder block in AHB mode. (**4-Byte align**)

NFIECC+000Ch NFIECC Encoder Idle Status Register NFIECC_ENCIDLE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ENC_IDLE**
Type																R
Reset																0
Timing																Immd

This register is for NFIECC Encoder idle status.

ENC_IDLE indicates the Encode block in idle state and ready for new message block.

- **0** means the Encode block is under working.
- **1** means the Encode block is in Idle state and available for new message block.

NFIECC+0010h NFIECC Parity0 Register NFIECC_ENCPAR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ENC_PAR0**															
Type	R															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ENC_PAR0**															
Type	R															
Reset	0															
Timing	Immd															

 © 2009 MediaTek Inc.

The register indicates the highest order of parity bits

ENC_PAR0 indicates the highest order of output parity bits and the bit 0 is the highest order of parity bit. The PAR0~PAR4 register is remain the last message block parity bits until ENC_EN is deasserted. The parity bits should append after main data by order of {PAR0[31:0], PAR1[31:0], PAR2[31:0], PAR3[31:0], PAR4[31:4], 4'b0}, The redundant bit of parity bit will be padded by 0.

NFIECC+0014h NFIECC Parity1 Register NFIECC_ENCPAR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ENC_PAR1															
Type	R															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENC_PAR1															
Type	R															
Reset	0															
Timing	Immd															

The register indicates the parity bits

ENC_PAR1 indicates the parity bits and the bit 0 is the highest order of parity bit.

NFIECC+0018h NFIECC Parity2 Register NFIECC_ENCPAR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ENC_PAR2															
Type	R															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENC_PAR2															
Type	R															
Reset	0															
Timing	Immd															

The register indicates the parity bits

ENC_PAR2 indicates the parity bits and the bit 0 is the highest order of parity bit.

NFIECC+001Ch NFIECC Parity3 Register NFIECC_ENCPAR3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ENC_PAR3															
Type	R															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENC_PAR3															
Type	R															
Reset	0															
Timing	Immd															

© 2009 MediaTek Inc.

The register indicates the parity bits

ENC_PAR3 indicates the parity bits and the bit 0 is the highest order of parity bit.

NFIECC+0020h NFIECC Parity4 Register NFIECC_ENCPAR4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**ENC_PAR4**											
Type					R											
Reset					0											
Timing					Immd											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ENC_PAR4**															
Type	R															
Reset	0															
Timing	Immd															

The register indicates the parity bits

ENC_PAR4 indicates the parity bits and the 31 is the highest order of parity bit.

NFIECC+0024h NFIECC Encoder Status Register NFIECC_ENCSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**COUNT_MS**													
Type			R													
Reset			0													
Timing			Immd													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT_PS**										**ENC_FSM**					
Type	R										R					
Reset	0										0					
Timing	Immed										Immed					

This register is for NFIECC Encoder status for SW polling.

ENC_FSM indicates encoder finite state machine state..

- **6'd0** IDLE
- **6'd1** WAITIN
- **6'd2** BUSY
- **6'd4** PAROUT

COUNT_PS indicates the parity bits that have **not** read out from NFI.

COUNT_MS indicates the remaining un-processing message bits.

NFIECC+0028h NFIECC Encoder IRQ enable Register NFIECC_ENCIRQEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ENC_IRQEN**
Type																R/W
Reset																0
Timing																Immd

This register is for software programmer to enable NFIECC IRQ signals (**ignore in NFI mode**)

ENC_IRQEN Encoder IRQ mask: triggered when Encoder operation is completed.

0 Disable

1 Enable

NFIECC+002Ch NFIECC Encoder IRQ status Register NFIECC_ENCIRQSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ENC_IRQSTA**
Type																RC
Reset																0
Timing																Immd

This register is for software programmer tracking NFIECC IRQ status. (**ignore in NFI mode**)

ENC_IRQSTA indicates interrupt status for Encoder processing.

0 No interrupt is generated.

1 An interrupt is pending and waiting for service. Active when Encoder processing is done.

NFIECC+0100h NFIECC Decoder Control Register NFIECC_DECCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**DEC_EN**
Type																R/W
Reset																0
Timing																Immd

This register is for Decoder control.

DEC_EN indicates the enable in NFI mode and start to work in AHB mode. In AHB mode, the decode-status FER and error number registers and error location registers will be reset to 0 when DEC_EN is deasserted.

0 means disable the Decode block.

1 means enable the Decode block. In AHB mode, the Decoder starts to fetch data when the register changes **from 0 to 1**. In NFI mode, the register enables the Decode block, and then the Decoder module waits start signal and data from NFI.

NFIECC+0104h NFIECC Decoder Configure Register NFIECC_DECCNFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEC_EMPTY_EN**			**DEC_CS**												
Type	R/W			R/W												
Reset	0			0												
Timing	Immd			Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name			DEC_CON									DEC_NFI_MODE		DEC_TNUM		
Type			R/W									R/W		R/W		
Reset			3									0		0		
Timing			Immd									Immd		Immd		

This register is for NFIECC configuration.

DEC_TNUM indicates the correct capability in one block size.

- **0** means the Decoder is capable of correct 4 bits in one block size.
- **1** means the Decoder is capable of correct 6 bits in one block size.
- **2** means the Decoder is capable of correct 8 bits in one block size.
- **3** means the NFIECC is capable of correct 10 bits in one block size.
- **4** means the NFIECC is capable of correct 12 bits in one block size.

DEC_NFI_MODE indicates the data source from access AHB bus or from NFI.

- **0** means input data from access AHB bus.
- **1** means input data from NFI module.

DEC_CON indicates the bypass configuration in decoding processor.

- **0** is reserved
- **1** means only active syndrome calculator for error detecting purpose. ECC reports **DONE** and **FER** status after syndrome calculator is done.
- **2** means error-correction module is bypassed for being aware of error location purpose. ECC reports **DONE**, **FER**, **EL** and **ERRNUM** status after Chien search is done.
- **3** means the ECC processor decoded data and auto-correction error data. The data address is signaled by DEC_DIADDR register in AHB mode and NFI_DIADDR in NFI mode. ECC reports **DONE**, **FER**, **EL** and **ERRNUM** status after error-correction is done.

DEC_CS indicates the total bit size of coded block **including protected data and parity bits**. The acceptable coded block size is 1~8191bits. If the coded block size is equal to zero, the decoder does nothing. The detail figure shows in Figure 2.

DEC_EMPTY_EN indicates the Decoder automatically detects the empty source data and by pass the auto-correction block (data are all equal to 1). **(ignore in AHB_mode)**

- **0** means disenable the detection of empty source data.
- **1** means enable the detection of empty source data.

NFIECC+0108h NFIECC Decoder DI Memory Address Register NFIECC_DECDIADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEC_DIADDR															
Type	R/W															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEC_DIADDR															
Type	R/W															
Reset	0															

Timing	Immd

The register indicates the data start address of input data to the Decoder AHB mode.

DEC_DIADDR indicates the memory address of input data to the Decoder block in AHB mode. **(4-Byte align)**.

NFIECC+010Ch NFIECC Decoder Idle Status Register NFIECC_DECIDLE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**DEC_IDLE**
Type																R
Reset																0
Timing																Immd

This register indicates the Decoder Idle status.

DEC_IDLE indicates the Decode block is in idle state and ready for new coded block.

0 means the Decode block is under working.

1 means the Decode block is in idle state and available for new coded block.

NFIECC+0110h NFIECC Decoder Found Error Status Register NFIECC_DECFER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FER7**	**FER6**	**FER5**	**FER4**	**FER3**	**FER2**	**FER1**	**FER0**
Type									R	R	R	R	R	R	R	R
Reset									0	0	0	0	0	0	0	0
Timing									Tsyn	Tsyn	Tsyn	Tsyn	Tsyn	Tsyn	Tsyn	Tsyn

This register is for NFIECC Decoder status.

FER*X* indicates the error found or not in the coded block. The FER numbered by NFI sector number in NFI mode, otherwise, in AHB mode, always use the FER0. The signal reset when DEC_EN is deasserted in both NFI and AHB mode.

0 means there is no error detected in the coded block.

1 means there is(are) error(s) detected in the coded block.

NFIECC+0114h NFIECC Decode Error Number Register NFIECC_DECENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ERRNUM7**				**ERRNUM6**				**ERRNUM5**				**ERRNUM4**			
Type	R				R				R				R			
Reset	0				0				0				0			
Timing	Tel				Tel				Tel				Tel			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ERRNUM3**				**ERRNUM2**				**ERRNUM1**				**ERRNUM0**			
Type	R				R				R				R			
Reset	0				0				0				0			
Timing	Tel				Tel				Tel				Tel			

The register indicates the error number of the coded block.

ERRNUMX indicates the error numbers of coded block in one start signal. **4'hf** means the error is uncorrectable. But ECC only can partially detect uncorrectable error.

NFIECC+0118h NFIECC Decoder Error Status Register NFIECC_DECDONE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DONE7**	**DONE6**	**DONE5**	**DONE4**	**DONE3**	**DONE2**	**DONE1**	**DONE0**
Type									R	R	R	R	R	R	R	R
Reset									0	0	0	0	0	0	0	0
Timing									Tdone	Tdone	Tdone	Tdone	Tdone	Tdone	Tdone	Tdone

This register is for NFIECC Decoder done status.

DONEX indicates the Decoding procedure is done.

0 means the Decode block is under working.

1 means the Decode block is finished. For different DEC_CON and EMPTY_EN in NFIECC_DECCNFG register, decoder done has different meaning. Detail of the definitions show in the Table 7.

	DEC_CON	NFI_MODE	EMPTY_EN	memo
Tdone=Tsyn	1	-	-	Syndrome stage detects there is error or not in main data and reports FER.
	1 or 2 or 3	-	-	Syndrome detects there is no error in main data, then done will be asserted.
	1 or 2 or 3	1	1	When NFI input an empty sector, ECC will automatically stop.
Tdone=Tel	2	-	-	Error location stage is finished and error location and error number is updated.
	3	-	-	When error number found in Chien search is over correct capability, ECC will automatically stop.
Tdone=Tcorr	3	-	-	ECC found error in coded block and finished all correction process.

Table 7 ***The ECC done signal asserts timing.***

NFIECC+011C NFIECC Decoder Error location0 Register NFIECC_DECEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**DEC_EL1**												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DEC_EL0**												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL0 indicates the error location 0 of the decoding result. The EL remains until the DEC_EN is deasserted to 0 in both AHB and NFI mode. When the error number is less than 12, error location registers will be filled from DEC_EL0, and the redundant register fields remain 0.

DEC_EL1 indicates the error location 1 of the decoding result.

NFIECC+0120h NFIECC Decoder Error Location1 Register NFIECC_DECEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**DEC_EL3**												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DEC_EL2**												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL2 indicates the error location 2 of the decoding result.

DEC_EL3 indicates the error location 3 of the decoding result.

NFIECC+0124h NFIECC Decoder Error Location2 Register NFIECC_DECEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**DEC_EL5**												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DEC_EL4**												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL4 indicates the error location 4 of the decoding result.

DEC_EL5 indicates the error location 5 of the decoding result.

NFIECC+0128h NFIECC Decoder Error Location3 Register NFIECC_DECEL3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**DEC_EL7**												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DEC_EL6**												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL6 indicates the error location 6 of the decoding result.
DEC_EL7 indicates the error location 7 of the decoding result.

NFIECC+012Ch NFIECC Decoder Error Location4 Register NFIECC_DECEL4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DEC_EL9												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DEC_EL8												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL8 indicates the error location 8 of the decoding result.
DEC_EL9 indicates the error location 9 of the decoding result.

NFIECC+0130h NFIECC Decoder Error Location5 Register NFIECC_DECEL5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DEC_EL11												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DEC_EL10												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_EL10 indicates the error location 8 of the decoding result.
DEC_EL11 indicates the error location 9 of the decoding result.

NFIECC+00134h NFIECC Decoder IRQ enable Register NFIECC_DECIRQEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DEC_IRQEN
Type																R/W
Reset																0
Timing																Immd

This register is for software programmer to enable NFIECC IRQ signals **(ignore in NFI mode)**

DEC_IRQEN Decoder IRQ mask: triggered when Decoder operation is completed.

0 Disable
1 Enable

NFIECC+0138h NFIECC Decoder IRQ status Register NFIECC_DECIRQSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**DEC_IRQSTA**
Type																RC
Reset																0
Timing																Immd

This register is for software programmer tracking NFIECC IRQ status. **(ignore in NFI mode)**

DEC_IRQSTA indicates Interrupt status for Decoder processing.
0 No interrupt is generated.
1 An interrupt is pending and waiting for service. Active when Decoder processing is done.

NFIECC+013Ch NFIECC FDM Register Address NFIECC_FDMADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FDM_ADDR**															
Type	R/W															
Reset	0															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FDM_ADDR**															
Type	R/W															
Reset	0															
Timing	Immd															

The register indicates the address of FDM data in NFI module.

FDM_ADDR indicates the APB register address of FDM data in NFI module.
In NFI mode, NFI_FDM_MODE*, and DEC_CON = 2, ECC will correct (**DEC_CS–PARITY**) location errors. ECC **assumes main data is 512 bytes** and correct the error location in the first 512 bytes using NFI_STR_ADDR and correct the remained errors using FDM_ADDR to find FDM data, which is in the NFI module, and will not correct parity data errors.
In NFI mode, NFI_FDM_MODE is **disable**, and DEC_CON = 2, ECC will correct all errors using NFI_STR_ADDR. But, if there is any FDM data is not protected by ECC, **ECC would not realize those data and might cause data being polluted.**

- * : NFI_FMD_MODE in NFI module is equal to AUTO_FMT_EN.

NFIECC+0140h NFIECC Decoder FSM NFIECC_DECFSM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**AUTOC_FSM**								**CHIEN_FSM**				
Type				R								R				

Reset				0								0				
Timing				Immd								Immd				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**BMA_FSM**							**SYN_FSM**					
Type				R							R					
Reset				0							0					
Timing				Immd							Immd					

The register indicates the finite state machine status of decoder.

SYN_FSM indicates the status of syndrome stage.

- **6'd0** IDLE
- **6'd1** WAITIN
- **6'd2** BUSY
- **6'd4** DONE

BMA_FSM indicates the status of BMA stage.

- **5'd0** IDLE
- **5'd1** BUSY
- **5'd2** DONE

CHIEN_FSM indicates the status of Chien search stage.

- **5'd0** IDLE
- **5'd1** BUSY
- **5'd2** DONE

AUTOC_FSM indicates the status of auto-correction stage.

- **5'd0** IDLE
- **5'd1** READ
- **5'd2** CHECK
- **5'd4** WRITE
- **5'd8** DONE

NFIECC+0144h NFIECC Syndrome Status Register NFIECC_SYNSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SYN_SNUM**						**DIBW**							**NFI_SEC_NUM**		
Type	0						R							R		
Reset	R						0							0		
Timing	Immd						Immd							Immd		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**NFI_STR_SET**		**SYN_COUNT_CS**													
Type	R		R													
Reset	0		0													
Timing	Immd		Immd													

This register is for NFIECC Syndrom status.

SYN_COUNT_CS indicates the remaining un-processing coded block bits.

NFI_STR_SET indicates the NFI_STR signal from NFI.
NFI_SEC_NUM indicates the sector number from NFI.
DIBW indicates input bandwidth.
SYN_SNUM indicates the sector number recorded by syndrome.

NFIECC+0148h NFIECC NFI input data Register NFIECC_DECNFIDI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	NFI_DI															
Type	0															
Reset	R															
Timing	Immd															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	NFI_DI															
Type	0															
Reset	R															
Timing	Immd															

This register is for checking NFI input data.

NFI_DI indicates the latest 4 byte input data from nfi.

NFIECC+014Ch NFIECC Syndrom Register NFIECC_SYN0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DEC_SYN3												
Type				R												
Reset				0												
Timing				Immd												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DEC_SYN1												
Type				R												
Reset				0												
Timing				Immd												

The register indicates the error location of the decoding result.

DEC_SYN1 informs the syndrome 1 from syndrome calculator.
DEC_SYN3 informs the syndrome 3 from syndrome calculator.

2.25.3 Timing Control Flow and Programming Sequence

This section lists the program sequences for ECC operations.

Caution: For MT6516, NFIECC module is in the AP site of MCUSYS, thus the address (ENC_DIADDR and DEC_DIADDR) should be assigned based on AP site MCU address.

2.25.3.1 Encoding in NFI mode

Caution: Before NFI address phase enable and configure ECC.

Configure	**Memo**
*NFIECC_ENCCNFG = 0x10400010;	//configure Encoder parameter in NFI mode.
*NFIECC_ENCCON = 0x1 ;	//enable Encoder.

while (NFI_STR==0x1) ;	//NFI_STR is happened in NFI address phase. NFI_STR is from NFI.
0 = *NFIECC_ENCIDLE ;	//It indicates the start is triggered and Encoder is in busy state.
while (*NFIECC_ENCIDLE==0x1) ;	//Wait all message data from NFI. After all data has input IDLE will be asserted.
parity = {*NFIECC_PAR0,*NFIECC_PAR1, *NFIECC_PAR2, *NFIECC_PAR3, *NFIECC_PAR4}	//If parity is necessary, Read out parity from APB register after IDLE=1.

2.25.3.2 Encoding in AHB mode

Configure	**Memo**
while (*NFIECC_ENCIDLE==1) ;	//polling IDLE signal until Encoder is available.
*NFIECC_ENCCNFG = 0x10400010;	//configure Encoder parameter in NFI mode.
*NFIECC_ENCIRQEN = 0x1;	//If IRQ is required when Encoder is done.
*NFIECC_ENCDIADDR= 0x10000000;	//Configure Data start address.
*NFIECC_ENCCON = 0x1 ;	//Encoder starts fetching data from ENCDIADDR.
0 = *NFIECC_ENCIDLE ;	//It indicates the start is triggered and Encoder is in busy state.
while (*NFIECC_ENCIDLE==0x1) ;	//After all data has fetched and encoded, IDLE will be asserted.
parity = {*NFIECC_PAR0,*NFIECC_PAR1, *NFIECC_PAR2, *NFIECC_PAR3, *NFIECC_PAR4}	// Read out parity from APB register after IDLE=1 and must append parity bits behind the original data for decoding.

2.25.3.3 Decoding in NFI mode

Caution: Before NFI address phase enable and configure ECC.

Caution: When NFI AUTO_FMT_EN=0, ECC will correct all errors (include parity bits) found in Chien search. Be careful of those FDM data that was not protected by ECC. Those data would not be realized by ECC module and might be polluted by ECC module.

Caution: ECC correct limitation is error_limit = error_correct_capability. If the error number(data error number + parity error number) is bigger than the error_limit, ECC might decode error.

Configure	**Memo**
*NFIECC_DECCNFG = 0x90743010;	//configure Decoder parameter in NFI mode.
*NFIECC_FDMADDR = 0x800320A0;	//configure FDM0 APB address in NFI mode into FDMADDR. (NFI_BASE_ADDR+FDM0_OFFSET_ADDR)
*NFIECC_DECCON = 0x1 ;	//enable Decoder.
for i = 1:8	// 8 is equal to NFI read sector number.
while (NFI_STR==0x1) ;	//NFI_STR is happened in NFI address phase. NFI_STR is from NFI.
0 = *NFIECC_DECIDLE ;	//It indicates the start is triggered and Decoder is in busy state.

while (*NFIECC_DECIDLE==0x1) ;	//Wait all message data from NFI. After all data has input IDLE will be asserted and FER will be reported.
end for	
while (*NFIECC_DECDONE==0xff)	//Decoder and correction processor is done.
ERR_NUM = *NFIECC_DECENUM	//Read Error number
for i = 0 : (ERR_NUM-1)	//Read Error Location
ErrorLocation[i] = *NFIECC_EL0+2i;	
end for	
*NFIECC_DECCON = 0x0	//Disable Decoder.

2.25.3.4 Decoding in AHB mode

Configure	**Memo**
while (*NFIECC_DECIDLE==1) ;	//polling IDLE signal until Decoder is available.
*NFIECC_DECCNFG = 0x90743010;	//configure Decoder parameter in NFI mode.
*NFIECC_DECIRQEN = 0x1;	//If IRQ is required when Encoder is done.
*NFIECC_DECDIADDR= 0x10000000;	//Configure Data start address.
*NFIECC_DECCON = 0x1 ;	//Encoder starts to fetch data from DECDIADDR.
0 = *NFIECC_DECIDLE ;	//It indicates the start is triggered and Decoder is in busy state.
while (*NFIECC_DECDONE==0x1)	//Decoder and correction processor is done.
ERR_NUM = *NFIECC_DECENUM	//Read Error number
for i = 0 : (ERR_NUM-1)	//Read Error Location
ErrorLocation[i] = *NFIECC_EL0+2i;	
end for	
*NFIECC_DECCON = 0x0	//Disable Decoder.

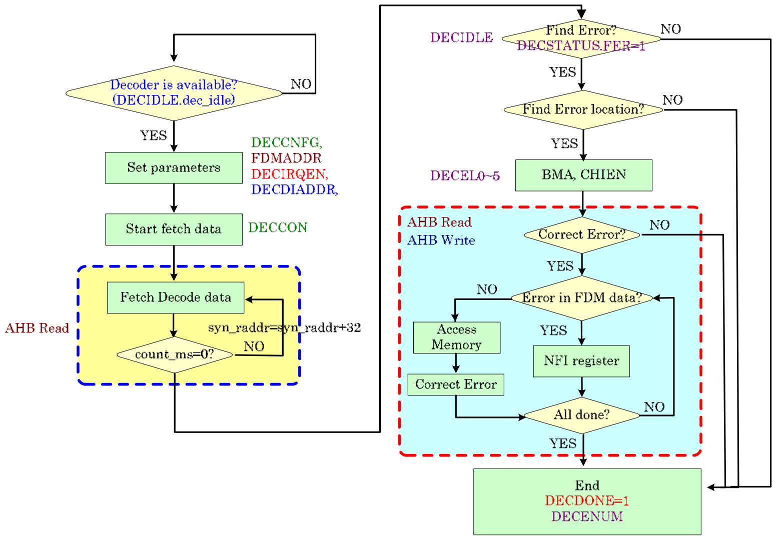


Figure 2 ***The recommendation flow of using Decoder.***

2.25.4 Control and Timing

2.25.4.1 Interface with NFI

The next waveform shows the data transfer protocol between ECC and NFI. The nfi_str is only triggered at beginning of a message block, and the input data is always companying with a nfi_divalid signal. ECC outputs nfi_di_rdy signal to indicate that the next data is ready for receiving.

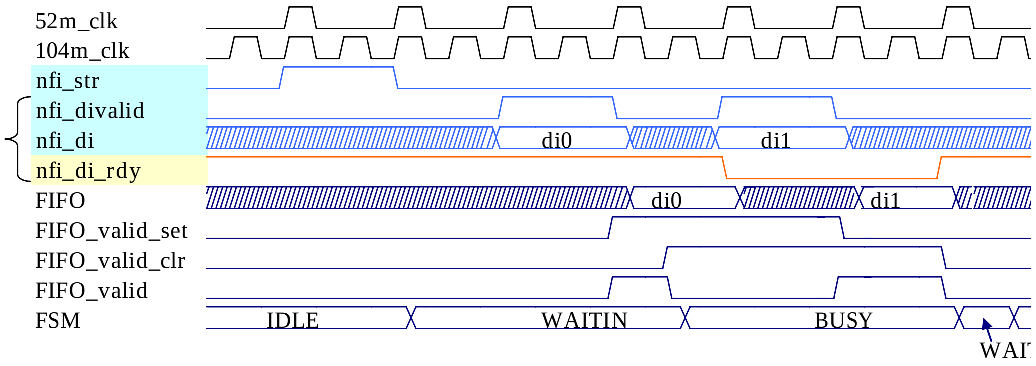


Figure 3 ***Timing Diagram of data input from NFI***

Besides data input from NFI, in encoding processing, ECC also needs to output parity data to NFI. When the total message blocks have been input, the nfi_par_rdy will pull high to indicate parity data is ready and ignore remaining input data. Figure 4 shows the timing diagram of output parity bits.

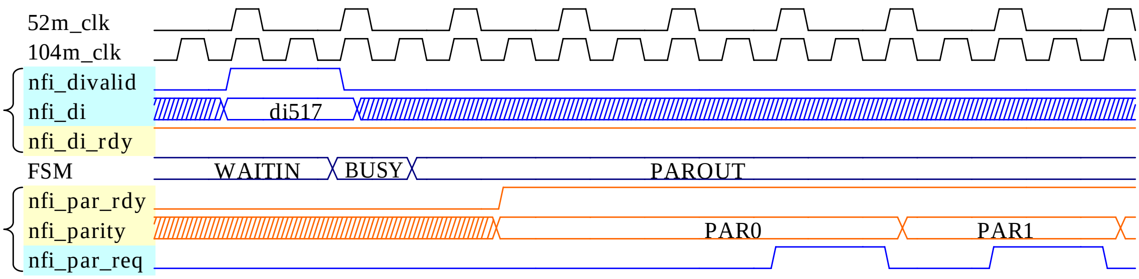


Figure 4 ***Timing diagram of parity output from ECC***

2.25.4.2 Interface with AHB

The sector describes the data transfer protocol between ECC and AHB arbiter. IDLE register indicates the module is ready or not to accept a new message block, thus AHB devices who want to use the encoder have to wait the register goes to high. When the enc_en or dec_en is asserted (from low to high), the module starts to capture the data from DIADDR. After all the data has been captured, ECC automatically goes back to IDLE state and pull up the IDLE signal. The parity bits will be stored in PA0~PA4.

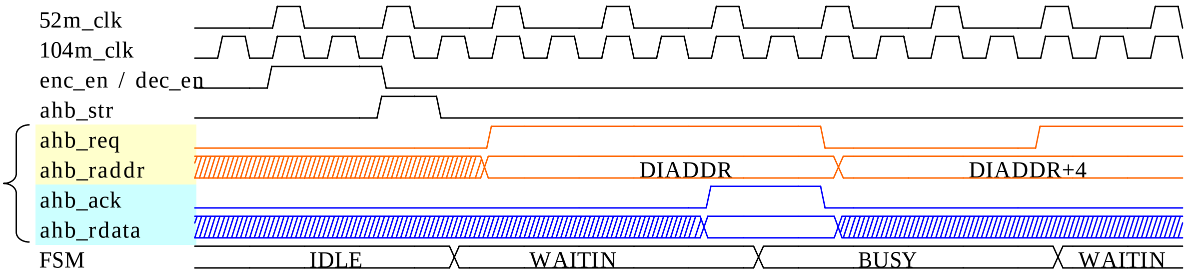


Figure 5 ***Timing Diagram of data input from AHB***

2.25.4.3 Encoding Architecture and FSM

The 2t consecutive powers of beta generates the generator polynomial which is defined as table below

t	*weight*	Generator Polynomial in hex
4	22	14523043ab86ab
6	44	7f3cc930e4f0dcb9b17d
8	49	115f914e07b0c138741c5c4fb23
10	65	65a4ef0d287c9a24ede0ab0157e0b37c9
12	68	1e4873256115a56784a6940a4c6e6d7e1205e051

Table 8 ***Generator polynomial in different correct capabilities.***

The encoding architecture is shown below:

© 2009 MediaTek Inc.

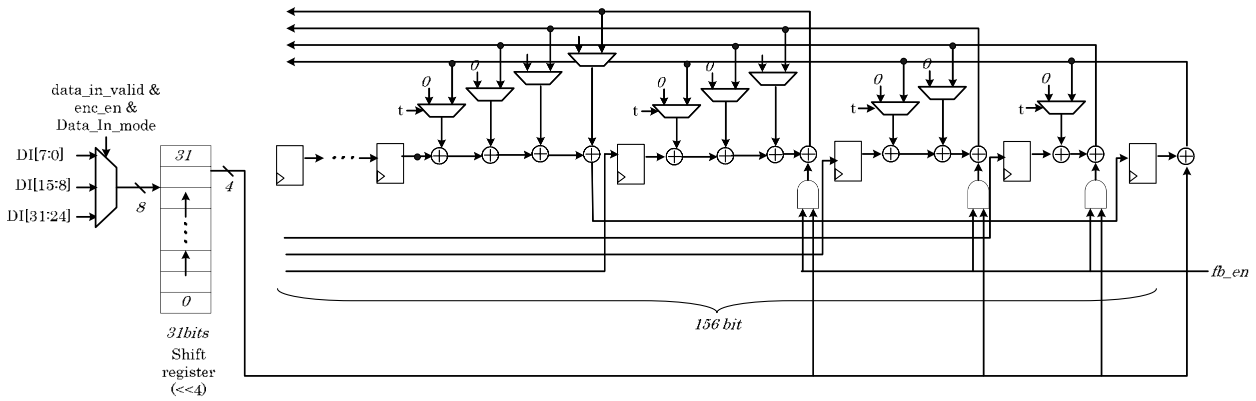


Figure 6 ***Encoder Architecture.***

The next Figure shows the finite state machine of Encoder. Encoder jumps from IDLE state to WAITIN state, after nfi_str signal is asserted. In WAITIN state, when enc_di_valid is asserted, Encoder jumps to processing state which is BUSY state. The Encoder goes to PAROUT state when all bits of message block are received.

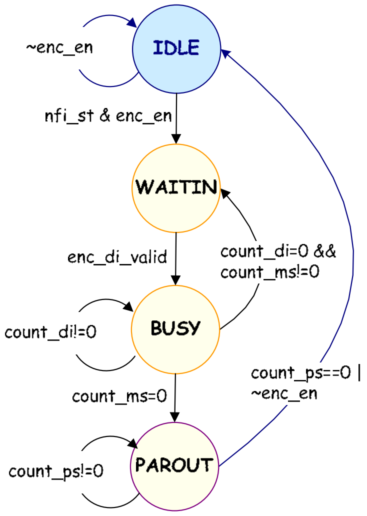


Figure 7 ***FSM of Encoder in NFI mod***

2.25.4.4 Decoding Architecture and FSM

The Decoding module can be easily separated to four main blocks, Syndrome calculation block, BMA block, Chien search block and Auto Error-Correction block. The protocol between the blocks is used REQ and ACK and shows in next figure. After syndromes have been calculated, the EL_REQ sent to BMA state and BMA state asserts EL_ACK if BMA is in IDLE state. The same way is in AC_REQ and AC_ACK.

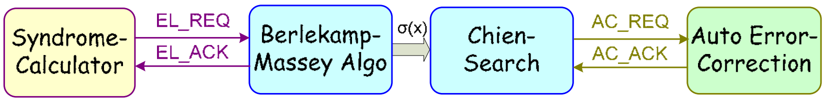


Figure 8 ***FSM of Decoder***

The BMA architecture is shown in next figure.

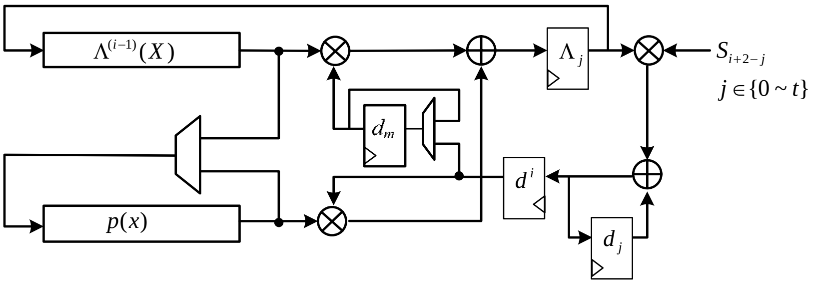


Figure 9 ***BMA block architecture.***

2.26 Reset Generation Unit (APRGU)

Figure 35 shows the reset scheme used in MT6516. MT6516 provides three kinds of resets: hardware reset, watchdog reset, and software reset. MT6516 provides 8 resets which can be manual reset by individual RGU_USRSTx control registers.

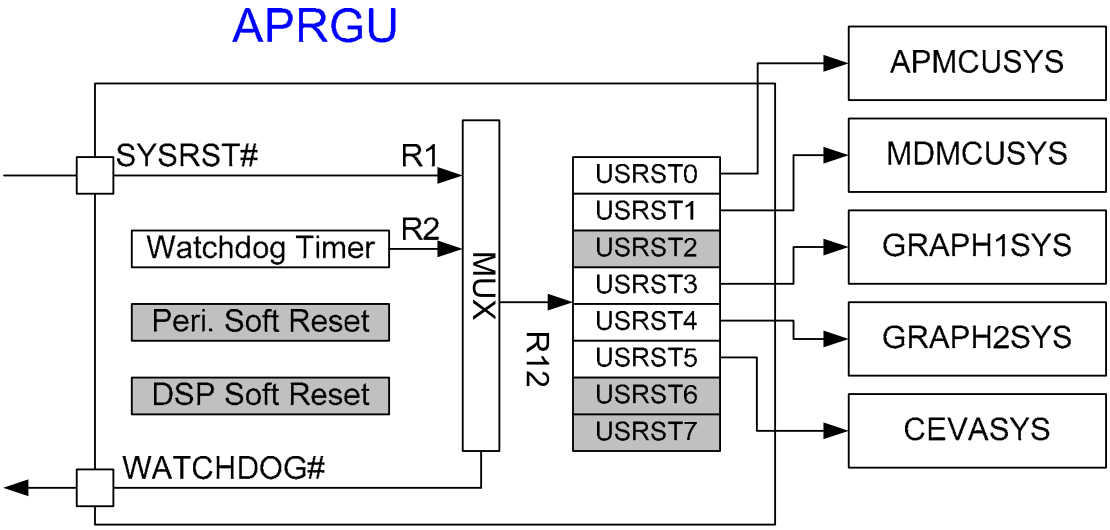


Figure 35 ***Reset Scheme Used in MT6516***

2.26.1 General Description

2.26.1.1 Hardware Reset

This reset is input through the SYSRST# pin, which is driven low during power-on. The hardware reset has a global effect on the chip: all digital and analog circuits are initialized, except the Real Time Clock module. The initial states of the MT6516 sub-blocks are as follows:

- All analog circuits are turned off.
- All PLLs are turned off and bypassed. The 13 MHz system clock is the default time base.

2.26.1.2 Watchdog Reset

A watchdog reset is generated when the Watchdog Timer expires: the MCU software failed to re-program the timer counter in time. This situation is typically induced by abnormal software execution, which can be aborted by a hardwired watchdog reset. Hardware blocks that are affected by the watchdog reset are:

- MCU subsystem,
- DSP subsystem, and
- External components (trigged by software).

2.26.1.3 Software Resets

Software resets are local reset signals that initialize specific hardware components. For example, if hardware failures are detected, the MCU or DSP software may write to software reset trigger registers to reset those specific hardware modules to their initial states.
The following modules have software resets.

- DSP Core
- DSP Coprocessors

2.26.2 Register Definitions

RGU +0000h Watchdog Timer Control Register WDT_MODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY[7:0]										WDTIN_DIS	AUTO-RESTART	IRQ	EXTEN	EXTPOL	ENABLE
Type	WO										R/W	R/W	R/W	R/W	R/W	R/W
Reset											1*	0	0	0	0	reset_en

ENABLE Enables the Watchdog Timer. The reset value depends on the ICORE: reset_en = ICORE.

- **0** Disables the Watchdog Timer.
- **1** Enables the Watchdog Timer.

EXTPOL Defines the polarity of the external watchdog pin.

- **0** Active low.
- 1 Active high.

EXTEN Specifies whether or not to generate an external watchdog reset signal.

- **0** The watchdog does not generate an external watchdog reset signal.
- **1** If the watchdog counter reaches zero, an external watchdog signal is generated.

IRQ Issues an interrupt instead of a Watchdog Timer reset. For debug purposes, RGU issues an interrupt to the MCU instead of resetting the system.

- **0** Disable.
- **1** Enable.

AUTO-RESTART Restarts the Watchdog Timer counter with the value of WDT_LENGTH while task ID is written into Software Debug Unit.

- **0** Disable. The counter restarts by writing KEY into the WDT_RESTART register.

 © 2009 MediaTek Inc.

1 Enable. The counter restarts by writing KEY into the WDT_RESTART register or by writing task ID into the software debug unit.

WDTIN_DIS If the other domain's watchdog affects the current domain's watchdog. ***WDTIN_DIS is only reset by external reset pin.**

0 This domain will be reset when other domain's watchdog is timeout.

1 This domain doesn't care about the other domain's watchdog.

KEY Write access is allowed if KEY=0x22.

RGU +0004h Watchdog Time-Out Interval Register WDT_LENGTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT[10:0]											KEY[4:0]				
Type	R/W											WO				
Reset	111_1111_1111b															

KEY Write access is allowed if KEY=08h.

TIMEOUT The counter is restarted with {TIMEOUT [10:0], 1_1111_1111b}. Thus the Watchdog Timer time-out period is a multiple of 512*T_{32k}=15.6ms.

RGU +0008h Watchdog Timer Restart Register WDT_RESTART

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY[15:0]															
Type	WO															
Reset																

KEY Restart the counter if KEY=1971h.

RGU +000Ch Watchdog Timer Status Register WDT_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WDT	SW_WDT														
Type	RO	RO														
Reset	0	0														

WDT Indicates the cause of the watchdog reset.

0 Reset not due to Watchdog Timer.

1 Reset because the Watchdog Timer time-out period expired.

SW_WDT Indicates if the watchdog was triggered by software.

0 Reset not due to software-triggered Watchdog Timer.

1 Reset due to software-triggered Watchdog Timer.

NOTE: A system reset does not affect this register. This bit is cleared when the WDT_MODE register is written.

RGU +0010h CPU Peripheral Software Reset Register SW_PERIPH_RSTN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RESERVED							
Type																

Reset								

RGU +0014h DSP Software Reset Register SW_DSP_RSTN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

RGU +0018h Watchdog Timer Reset Signal Duration Register WDT_RSTINTERVAL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**LENGTH[11:0]**											
Type					R/W											
Reset					FFFh											

LENGTH This register indicates the reset duration when Watchdog Timer times out. However, if the WDT_MODE register IRQ bit is set to 1, an interrupt is issued instead of a reset.

RGU+001Ch Watchdog Timer Software Reset Register WDT_SWRST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[15:0]**															
Type	WO															
Reset																

Software-triggered Watchdog Timer reset. If the register content matches the KEY, a watchdog reset is issued. However, if the WDT_MODE register IRQ bit is set to 1, an interrupt is issued instead of a reset.

KEY 1209h

RGU+0020h RGU user-defined reset 0 RGU_USRST0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST0[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0024h RGU user-defined reset 1 RGU_USRST1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST1[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0028h RGU user-defined reset 2 RGU_USRST2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST2[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+002ch RGU user-defined reset 3 RGU_USRST3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST3[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0030h RGU user-defined reset 4 RGU_USRST4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST4[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0034h RGU user-defined reset 5 RGU_USRST5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST5[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0038h RGU user-defined reset 6 RGU_USRST6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST6[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+003ch RGU user-defined reset 7 RGU_USRST7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST7[7:0]**							
Type	WO								R/W							
Reset									0							

KEY Write access is allowed if KEY=0xbb.

User-defined resets can trigger individual resets derived by MCU. When the USRSTx[7:0] is non-zero, the corresponding reset will be pull low. (All resets generated by RGU are active low).

USRSTx[6:0] is for a period of reset. It will decrease 1 per system clock when USRSTx[6:0] is not equal to 0. It is suitable for a predefined period reset.

USRSTx[7] is for a manual reset. It is only changed by MCU and suitable for a manual reset fully controlled by MCU.

Generally speaking, USRSTx[6:0] and USRSTx[7] will not be non-zero at the same time.

Example1:

0th cycle: USRSTx = 0x0. RESETx=1. MCU writes USRSTx = 0x3.

1st cycle: USRSTx = 0x3. RESETx=0.

2nd cycle: USRSTx = 0x2. RESETx=0.

3rd cycle: USRSTx = 0x1. RESETx=0.

4th cycle: USRSTx = 0x0. RESETx=1.

Example2:

0th cycle: USRSTx = 0x0. RESETx=1. MCU writes USRSTx = 0x80.

1st cycle: USRSTx = 0x80. RESETx=0.

2nd cycle: USRSTx = 0x80. RESETx=0.

~

Nth cycle: USRSTx = 0x80. RESETx=0. MCU writes USRSTx = 0x0.

N+1th cycle: USRSTx = 0x0. RESETx=1.

2.27 SIM Interface

The MT6516 contains two dedicated smart card interfaces to allow the MCU to access the two SIM cards. Each interface can operate via 5 terminals. As shown is the Figure 1, SIMVCC, SIMSEL, SIMRST, SIMCLK and SIMDATA are for one SIM interface, while SIM2VCC, SIM2SEL, SIM2RST, SIM2CLK and SIM2DATA are for the other one.

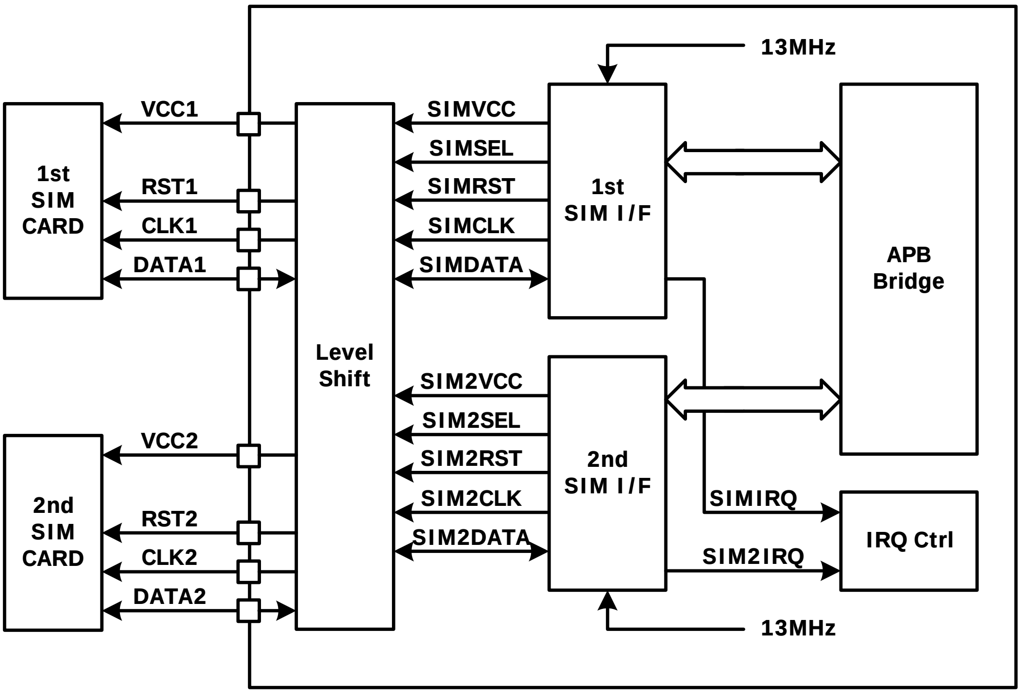


Figure 36 ***SIM Interface Block Diagram***

 © 2009 MediaTek Inc.

The functions of the two SIM interfaces are identical; therefore, only first SIM interface will be described in this document. The SIMVCC is used to control the external voltage supply to the SIM card and SIMSEL determines the regulated smart card supply voltage. SIMRST is used as the SIM card reset signal. Besides, SIMDATA and SIMCLK are used for data exchange purpose.

Basically, the SIM interface acts as a half duplex asynchronous communication port and its data format is composed of ten consecutive bits: a start bit in state Low, eight information bits, and a tenth bit used for parity checking. The data format can be divided into two modes as follows:

Direct Convention Mode (ODD=SDIR=SINV=0)

SB D0 D1 D2 D3 D4 D5 D6 D7 PB

SB: Start Bit (in state Low)

Dx: Data Byte (LSB is first and logic level ONE is in state High)

PB: Even Parity Check Bit

Inverse Convention Mode (ODD=SDIR=SINV=1)

SB N7 N6 N5 N4 N3 N2 N1 N0 PB

SB: Start Bit (in state Low)

Nx**:** Data Byte (MSB is first and logic level ONE is in state Low)

PB: Odd Parity Check Bit

If the receiver gets a wrong parity bit, it will respond by pulling the SIMDATA Low to inform the transmitter and the transmitter will retransmit the character.

When the receiver is a SIM Card, the error response starts 0.5 bits after the PB and it may last for 1~2 bit periods.

When the receiver is the SIM interface, the error response starts 0.5 bits after the PB and lasts for 1.5 bit period.

When the SIM interface is the transmitter, it will take totally 14 bits guard period whether the error response appears. If the receiver shows the error response, the SIM interface will retransmit the previous character again else it will transmit the next character.

© 2009 MediaTek Inc.

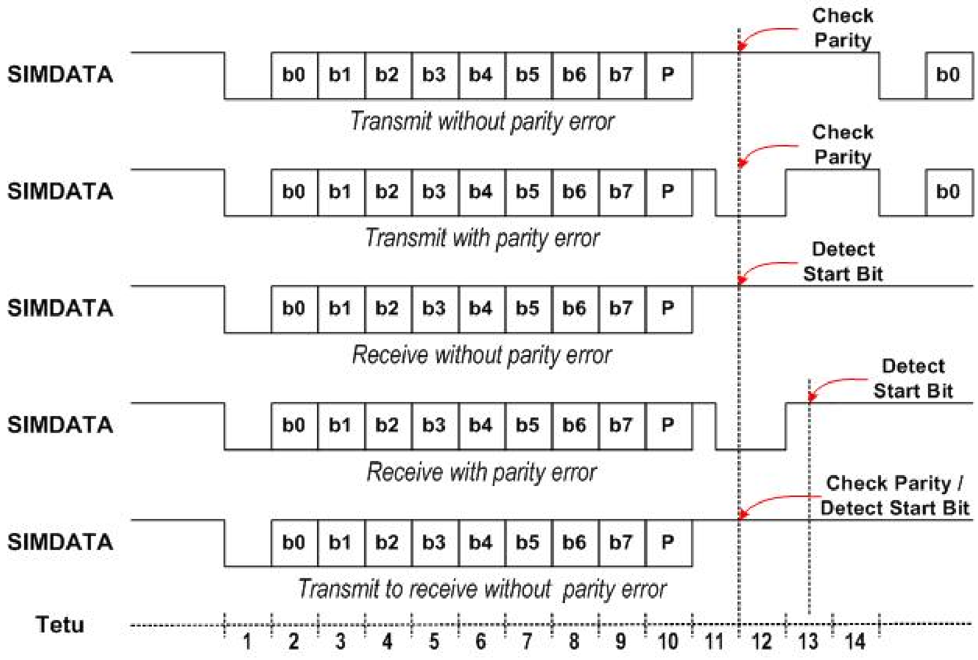


Figure 37 ***SIM Interface Timing Diagram***

2.27.1 Register Definitions

For MCU to control two SIM card interface, all registers are duplicated to two copies but with different base address. In the following, n = " " is for 1st SIM card interface, while n=2 is for 2nd SIM card interface. For example, address SIM+0000h is mapped to SIM_SIM_CONT register, while address SIM2+0000h is mapped to SIM2_SIM_CONT register.

SIMn+0000h SIM module control register SIMN_SIM_CONT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														WRST	CSTOP	SIMON
Type														W	R/W	R/W
Reset														0	0	0

SIMON SIM card power-up/power-down control

0 An 1-to-0 change will start the card deactivation sequence

1 A 0-to-1 change will start the card activation sequence

CSTOP Enable clock stop mode. Together with CPOL in SIM_CONF register, it determines the polarity of the SIMCLK in this mode.

0 Enable the SIMCLK output.

1 Disable the SIMCLK output

WRST SIM card warm reset control

SIMn+0004h SIM module configuration register SIMN_SIM_CONF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name						HFEN	T0EN	T1EN	TOUT	SIMSEL	ODD	SDIR	SINV	CPOL	TXACK	RXACK
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0

RXACK SIM card reception error handshake control
- **0** Disable character receipt handshaking
- **1** Enable character receipt handshaking

TXACK SIM card transmission error handshake control
- **0** Disable character transmission handshaking
- **1** Enable character transmission handshaking

CPOL SIMCLK polarity control in clock stop mode
- **0** Make SIMCLK stop in LOW level
- **1** Make SIMCLK stop in HIGH level

SINV Data invert mode
- **0** Not invert the transmitted and received data, data logic ONE is in high state
- **1** Invert the transmitted and received data, data logic ONE is in low state

SDIR Data Transfer Direction
- **0** LSB is transmitted and received first
- **1** MSB is transmitted and received first

ODD Select odd or even parity
- **0** Even parity
- **1** Odd parity

SIMSEL SIM card supply voltage select
- **0** SIMSEL pin is set to LOW level, 1.8V
- **1** SIMSEL pin is set to HIGH level, 3V

TOUT SIM work waiting time counter control
- **0** Disable Time-Out counter
- **1** Enable Time-Out counter

T1EN T=1 protocol controller control
- **0** Disable T=1 protocol controller
- **1** Enable T=1 protocol controller

T0EN T=0 protocol controller control
- **0** Disable T=0 protocol controller
- **1** Enable T=0 protocol controller

HFEN Hardware flow control
- **0** Disable hardware flow control
- **1** Enable hardware flow control

SIMn +0008h SIM Baud Rate Register SIMN_SIM_BRR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						ETU[8:0]									SIMCLK[1:0]	
Type						R/W									R/W	
Reset						372d									01	

SIMCLK Set SIMCLK frequency

- **00** 13/2 MHz
- **01** 13/4 MHz
- **10** 13/8 MHz
- 11 13/12 MHz

ETU Determines the duration of elementary time unit in unit of SIMCLK

SIMn +0010h SIM interrupt enable register SIMN_SIM_IRQEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						EDCERR	T1END	RXERR	T0END	SIMOFF	ATRERR	TXERR	TOUT	OVRUN	RXTIDE	TXTIDE
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0

For all these bits

- **0** Interrupt is disabled
- **1** Interrupt is enabled

SIMn +0014h SIM module status register SIMN_SIM_STS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						EDCERR	T1END	RXERR	T0END	SIMOFF	ATRERR	TXERR	TOUT	OVRUN	RXTIDE	TXTIDE
Type						R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R	R
Reset						—	—	—	—	—	—	—	—	—	—	—

TXTIDE The interrupt occurs when number of transmitted data in the FIFO is less than transmitted tide.
RXTIDE The interrupt occurs when number of received data in the FIFO is less than received tide.
OVRUN Transmit/Receive FIFO overflow interrupt occurred
TOUT Between characters timeout interrupt occurred
TXERR Character transmission error interrupt occurred
ATRERR ATR start time-out interrupt occurred
SIMOFF Card deactivation complete interrupt occurred
T0END Data Transfer handled by T=0 Controller completed interrupt occurred
RXERR Character reception error interrupt occurred
T1END Data Transfer handled by T=1 Controller completed interrupt occurred
EDCERR T=1 Controller CRC error occurred

SIMn +0020h SIM retry limit register SIMN_SIM_RETRY

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TXRETRY								RXRETRY		
Type						R/W								R/W		
Reset						3h								3h		

RXRETRY Specify the maximum numbers of receive retries that are allowed when parity error has occurred.
TXRETRY Specify the maximum numbers of transmit retries that are allowed when parity error has occurred.

 © 2009 MediaTek Inc.

SIMn +0024h SIM FIFO tide mark register SIMN_SIM_TIDE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TXTIDE[3:0]								RXTIDE[3:0]			
Type					R/W								R/W			
Reset					0h								0h			

RXTIDE Trigger point of RXTIDE interrupt

TXTIDE Trigger point of TXTIDE interrupt

SIMn +0030h Data register used as Tx/Rx Data Register SIMN_SIM_DATA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA[7:0]							
Type									R/W							
Reset									—							

DATA Eight data digits. These correspond to the character being read or written

SIMn +0034h SIM FIFO count register SIMN_SIM_COUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												COUNT[4:0]				
Type												R/W				
Reset												0h				

COUNT The number of characters in the SIM FIFO when read, and flushes when written.

SIMn +0040h SIM activation time register SIMN_SIM_ATIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ATIME[15:0]															
Type	R/W															
Reset	AFC7h															

ATIME The register defines the duration, in SIM clock cycles, of the time taken for each of the three stages of the card activation process, from SIMON transits to high to turn on VCC, from turn on VCC to pull DATA high and then from pull DATA high to turn on CLK.

SIMn +0044h SIM deactivation time register SIMN_SIM_DTIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DTIME[11:0]											
Type					R/W											
Reset					3E7h											

DTIME The register defines the duration, in 13MHz clock cycles, of the time taken for each of the three stages of the card deactivation sequence, from pull RST low to turn of CLK, from turn off CLK to pull DATA low, from pull DATA low to turn off to turn off VCC.

SIMn +0048h Character to character waiting time register SIMN_SIM_WTIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WTIME[15:0]															
Type	R/W															

© 2009 MediaTek Inc.

Reset	983h															

WTIME Maximum interval between the leading edge of two consecutive characters in 4 ETU unit

SIMn +004Ch Block to block guard time register SIMN_SIM_GTIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GTIME			
Type													R/W			
Reset													10d			

GTIME Minimum interval between the leading edge of two consecutive characters sent in opposite directions in ETU unit

SIMn +0050h Block to error signal time register SIMN_SIM_ETIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ETIME					
Type											R/W					
Reset											15d					

ETIME The register defines the interval, in 1/16 ETU unit, between the end of transmitted parity bit and time to check parity error signal sent from SIM card.

SIMn +0060h SIM command header register: INS SIMN_SIM_INS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								INSD	SIMINS[7:0]							
Type								R/W	R/W							
Reset								0h	0h							

SIMINS This field should be identical to the INS instruction code. When writing to this register, the T=0 controller will be activated and data transfer will be initiated.

INSD [Description for this register field]

- **0** T=0 controller receives data from the SIM card
- **1** T=0 controller sends data to the SIM card

SIMn +0064h SIM command header register: P3 SIMN_SIM_P3 (ICC_LEN)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SIMP3[8:0]								
Type								R/W								
Reset								0h								

SIMP3 This field should be identical to the P3 instruction code. It should be written prior to the SIM_INS register. While the data transfer is going on, this field shows the no. of the remaining data to be sent or to be received

SIMn +0068h SIM procedure byte register: SW1 SIMN_SIM_SW1 (ICC_LEN)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SIMSW1[7:0]							
Type									R							

Reset									0h							

SIMSW1 This field holds the last received procedure byte for debug purpose. When the T0END interrupt occurred, it keeps the SW1 procedure byte.

SIMn +006Ch SIM procedure byte register: SW2 SIMN_SIM_SW2 (ICC_EDC)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SIMSW2[7:0]							
Type									R							
Reset									0h							

SIMSW2 This field holds the SW2 procedure byte

2.27.2 SIM Card Insertion and Removal

The detection of physical connection to the SIM card and card removal is done by the external interrupt controller or by GPIO.

2.27.3 Card Activation and Deactivation

The card activation and deactivation sequence both are controlled by H/W. The MCU initiates the activation sequence by writing a “1” to bit 0 of the SIM_CON register, and then the interface performs the following activation sequence:

- Assert SIMRST LOW
- Set SIMVCC at HIGH level and SIMDATA in reception mode
- Enable SIMCLK clock
- De-assert SIMRST HIGH (required if it belongs to active low reset SIM card)

The final step in a typical card session is contact deactivation in order that the card is not electrically damaged. The deactivation sequence is initiated by writing a “0” to bit 0 of the SIM_CONT register, and then the interface performs the following deactivation sequence:

- Assert SIMRST LOW
- Set SCIMCLK at LOW level
- Set SIMDATA at LOW level
- Set SIMVCC at LOW level

2.27.4 Answer to Reset Sequence

After card activation, a reset operation results in an answer from the card consisting of the initial character TS, followed by at most 32 characters. The initial character TS provides a bit synchronization sequence and defines the conventions to interpret data bytes in all subsequent characters.

On reception of the first character, TS, MCU should read this character, establish the respective required convention and reprogram the related registers. These processes should be completed prior to the completion of reception of the next character. And then, the remainder of the ATR sequence is received, read via the SIM_DATA in the selected convention and interpreted by the S/W.

The timing requirement and procedures for ATR sequence are handled by H/W and shall meet the requirement of ISO 7816-3 as shown in **Figure 38**.

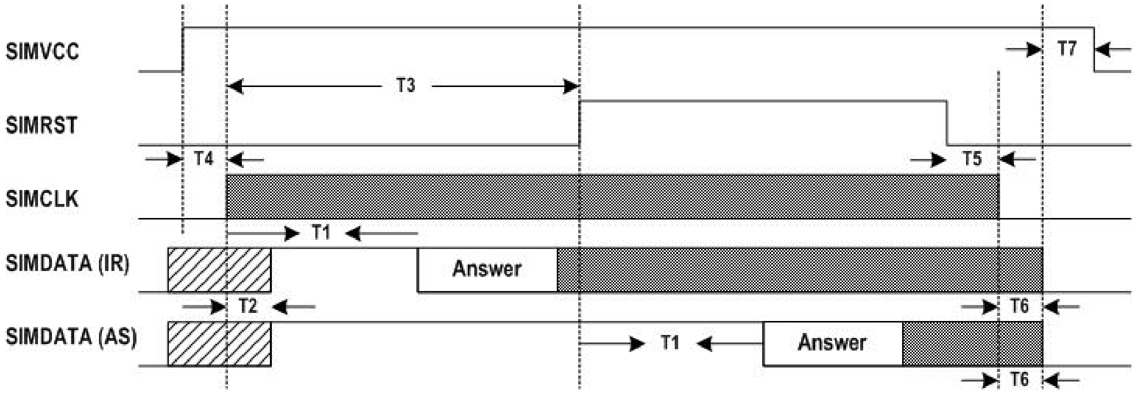


Figure 38 ***Answer to Reset Sequence***

Time	Value	Comment
T1	> 400 SIMCLK	SIMCLK start to ATR appear
T2	< 200 SIMCLK	SIMCLK start to SIMDATA in reception mode
T3	> 40000 SIMCLK	SIMCLK start to SIMRST High
T4	—	SIMVCC High to SIMCLK start
T5	—	SIMRST Low to SIMCLK stop
T6	—	SIMCLK stop to SIMDATA Low
T7	—	SIMDATA Low to SIMVCC Low

Table 46 ***Answer to Reset Sequence Time-Out Condition***

2.27.5 SIM Data Transfer

Two transfer modes are provided, either in software controlled byte by byte fashion or in a block fashion using T=0 controller and DMA controller. In both modes, the time-out counter could be enabled to monitor the elapsed time between two consecutive bytes.

1.1.1.1. Byte Transfer Mode

This mode is used during ATR and PPS procedure. In this mode, the SIM interface only ensures error free character transmission and reception.

Receiving Character

Upon detection of the start-bit sent by SIM card, the interface transforms into reception mode and the following bits are shifted into an internal register. If no parity error is detected or character-receive handshaking is disabled, the received-character is written into the SIM FIFO and the SIM_COUNT register is increased by one. Otherwise, the SIMDATA line is held low at 0.5 etu after detecting the parity error for 1.5 etus, and the character is re-received. If a character fails to be received correctly for the RXRETRY times, the receive-handshaking is aborted and the last-received character is written into the SIM FIFO, the SIM_COUNT is increased by one and the RXERR interrupt is generated

When the number of characters held in the receive FIFO exceeds the level defined in the SIM_TIDE register, a RXTIDE interrupt is generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register and writing to this register will flush the SIM FIFO.

Sending Character

Characters that are to be sent to the card are first written into the SIM FIFO and then automatically transmitted to the card at timed intervals. If character-transmit handshaking is enabled, the SIMDATA line is sampled at 1 etu after the parity bit. If the card indicates that it did not receive the character correctly, the character is retransmitted a maximum of TXRETRY times before a TXERR interrupt is generated and the transmission is aborted. Otherwise, the succeeding byte in the SIM FIFO is transmitted.

If a character fails to be transmitted and a TXERR interrupt is generated, the interface needs to be reset by flushing the SIM FIFO before any subsequent transmit or receive operation.

When the number of characters held in the SIM FIFO falls below the level defined in the SIM_TIDE register, a TXTIDE interrupt is generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register and writing to this register will flush the SIM FIFO.

1.1.1.2. Block Transfer Mode

Basically, the SIM interface is designed to work in conjunction with the T=0 protocol controller and the DMA controller during non-ATR and non-PPS phase, though it is still possible for software to service the data transfer manually like in byte transfer mode if necessary and thus the T=0 protocol should be controlled by software.

The T=0 controller is accessed via four registers representing the instruction header bytes INS and P3, and the procedure bytes SW1 and SW2. These registers are:

SIM_INS, SIM_P3

SIM_SW1, SIM_SW2

During characters transfer, SIM_P3 holds the number of characters to be sent or to be received and SIM_SW1 holds the last received procedure byte including NULL, ACK, NACK and SW1 for debug purpose.

Data Receive Instruction

Data Receive Instructions receive data from the SIM card. It is instantiated as the following procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in SIM_CONF register
2. Program the SIM_TIDE register to 0x0000 (TXTIDE = 0, RXTIDE = 0)
3. Program the SIM_IRQEN to 0x019C (Enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts)
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO
5. Program the DMA controller :
   DMA*n*_MSBSRC and DMA*n*_LSBSRC : address of SIM_DATA register
   DMA*n*_MSBDST and DMA*n*_LSBDST : memory address reserved to store the received characters
   DMA*n*_COUNT : identical to P3 or 256 (if P3 == 0)
   DMA*n*_CON : 0x0078
6. Write P3 into SIM_P3 register and then INS into SIM_INS register (Data Transfer is initiated now)
7. Enable the Time-out counter by setting the TOUT bit to 1 in SIM_CONF register
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register to

Upon completion of the Data Receive Instruction, T0END interrupt will be generated and then the Time-out counter should be disabled by setting the TOUT bit back to 0 in SIM_CONF register.

 © 2009 MediaTek Inc.

If error occurs during data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activated prior subsequent operations.

Data Send Instruction

Data Send Instructions send data to the SIM card. It is instantiated as the following procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in SIM_CONF register
2. Program the SIM_TIDE register to 0x0100 (TXTIDE = 1, RXTIDE = 0)
3. Program the SIM_IRQEN to 0x019C (Enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts)
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO
5. Program the DMA controller :
   DMA*n*_MSBSRC and DMA*n*_LSBSRC : memory address reserved to store the transmitted characters
   DMA*n*_MSBDST and DMA*n*_LSBDST : address of SIM_DATA register
   DMA*n*_COUNT : identical to P3
   DMA*n*_CON : 0x0074
6. Write P3 into SIM_P3 register and then (0x0100 | INS) into SIM_INS register (Data Transfer is initiated now)
7. Enable the Time-out counter by setting the TOUT bit to 1 in SIM_CONF register
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register

Upon completion of the Data Send Instruction, T0END interrupt will be generated and then the Time-out counter should be disabled by setting the TOUT bit back to 0 in SIM_CONF register.

If error occurs during data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activated prior subsequent operations.

 © 2009 MediaTek Inc.

2.28 Slow Clocking Unit for AP side

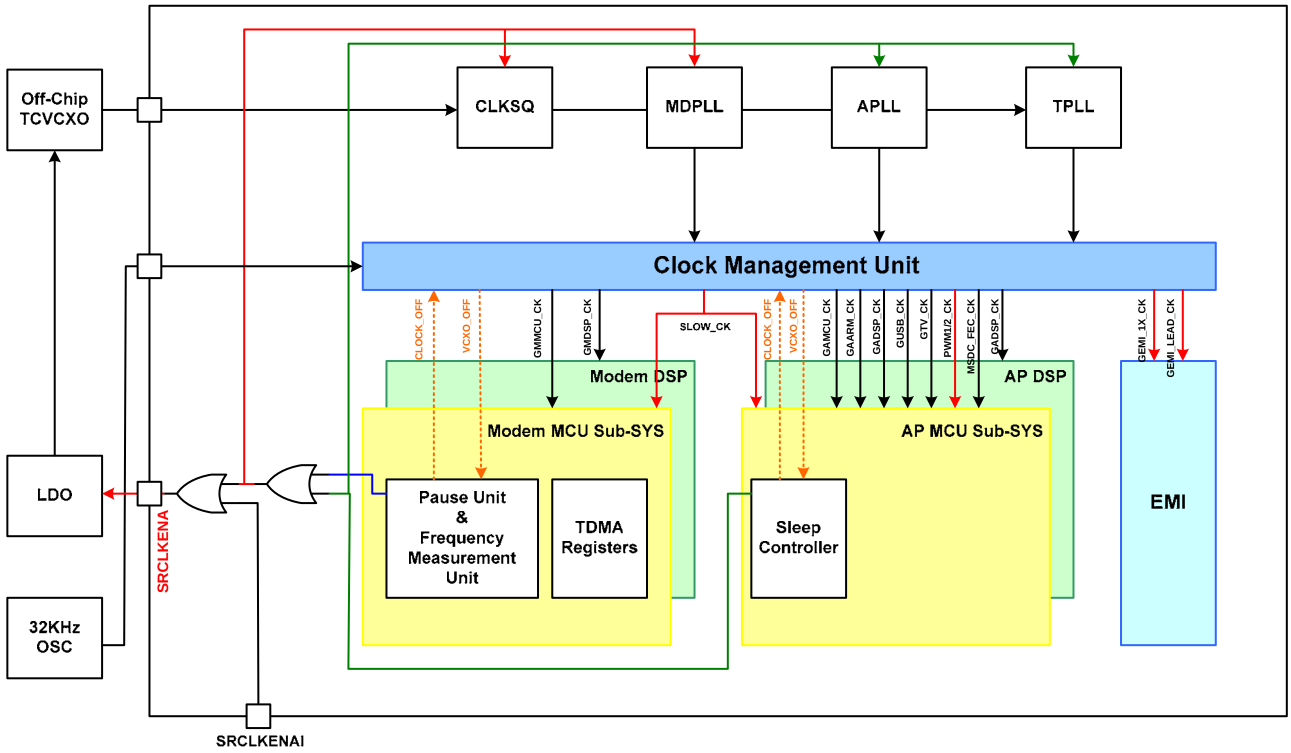


Figure 39 ***The block diagram of the slow clocking unit***

The slow clocking unit is provided to maintain the synchronization to a 32KHz crystal oscillator while the 13MHz reference clock is switched off. As shown in Figure 39, this unit is composed of pause unit, and clock management unit.

The pause unit is used to initiate and terminate the pause mode procedure and it also works as a coarse time-base during the pause period.

The clock management unit is used to control the system clock while switching between the normal mode and the pause mode. SRCLKENA is used to turn on/off the clock squarer, DSP PLL and off-chip TCVCXO. CLOCK_OFF signal is used for gating the main MCU and DSP clock, and VCXO_OFF is used as the acknowledgement signal of the CLOCK_OFF request.

2.28.1 Register Definitions

APSLP +0218h **Slow clocking unit control register** **AP_CON**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															PAUSE_START	
Type															W	

 © 2009 MediaTek Inc.

Reset															0	

PAUSE_START Initiate the pause mode procedure at the next timer wrap position. In order for safe ARM9 and bus power-down, CP15 instruction must be executed before triggering PAUSE_START

APSLP +021Ch Slow clocking unit status register AP_STA

Bit	15	14	13	12	11	10	9	8
Name								**PAUSE_ABORT**
Type								R.C
Bit	7	6	5	4	3	2	1	0
Name	**SETTLE_CPL**	**PAUSE_CPL**	**PAUSE_INT**	**PAUSE_RQST**				
Type	R.C	R.C	R.C	R.C				

PAUSE_RQST Pause mode procedure is requested
PAUSE_INT Asynchronous wake up from pause mode
PAUSE_CPL Pause period is completed
SETTLE_CPL Settling period is completed
PAUSE_ABORT Pause mode is aborted because of the reception of interrupt prior to entering pause mode

APSLP +022Ch Slow clocking unit configuration register AP_CNF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**MD2AP_CCIF**	**LOWBAT**	**TP**	**GPT**	**MSDC**	**RTC**	**EINT**	**KP**	**SM**	~~**SPARE**~~
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	~~R/W~~
Reset							0	0	0	0	0	0	0	0	1	~~1~~

SM Enable interrupt generation upon completion of pause mode procedure
KP Enable asynchronous wake-up from pause mode by key press
EINT Enable asynchronous wake-up from pause mode by external interrupt
RTC Enable asynchronous wake-up from pause mode by real time clock interrupt
MSDC Enable asynchronous wake-up from pause mode by memory card insertion interrupt
GPT Enable asynchronous wake-up from pause mode by general-purpose timer interrupt
TP Enable asynchronous wake-up from pause mode by touch panel interrupt
LOWBAT Enable asynchronous wake-up from pause mode by low-battery measurement from AUXADC
MD2AP_CCIF Enable asynchronous wake-up from pause mode by CCIF interface coming from MD side

APSLP +0228h Turn on pause-complete function AP_WAKE_PLL_SETTING

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**WITHOUT_PAUSE_CPL**														

Type		R/W														
Reset		0														

WITHOUT_PAUSE_CPL Turn on pause-complete function for AP sleep-controller

- **1** Turn off pause-complete, the AP sleep-controller keeps sleeping until one of the permitted interrupt sources of AP_CNF asserts
- **0** Turn on pause-complete, the AP sleep-controller automatically wake-up according to AP_PAUSE_M and AP_PAUSE_L, at most the AP sleep-controller can sleep by 16ms. After the counter expired, the controller wake-up and enters the settling stage, defined by AP_CLK_SETTLE.

Address	Type	Width	Reset Value	Name	Description
+0200h	R/W	[2:0]	—	AP_PAUSE_M	MSB of pause duration, in unit of 32kHz
+0204h	R/W	[15:0]	—	AP_PAUSE_L	16 LSB of pause duration, in unit of 32kHz
+0208h	R/W	[13:0]	—	AP_CLK_SETTLE	Off-chip VCXO settling duration, in unit of 32kHz
+020Ch	R	[2:0]	—	AP_FINAL_PAUSE_M	MSB of final pause count
+0210h	R	[15:0]	—	AP_FINAL_PAUSE_L	16 LSB of final pause count
+0218h	W	[1:0]	0x0000	AP_CON	SM control register
+021Ch	R	[8:4,1:0]	0x0000	AP_STA	SM status register
+022Ch	R/W	[7:0]	0x0003	AP_CNF	SM configuration register
+0230h	R	[7:0]	0x0000	RTCCOUNT_M	MSB of RTC count
+0234h	R	[15:0]	0x0000	RTCCOUNT_L	16 LSB of RTC count
+0238h	R/W	[15:0]	0x8020	WAKE_APLL_SETTING	Only bit 14 is valid. Used to indicate whether the controller wakes up by itself

2.29 UART

2.29.1 General Description

The baseband chipset houses three UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external devices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from five to eight bits**, **an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided. The UART also includes two DMA handshake lines, used to indicate when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note: The UART has been designed so that all internal operations are synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and the industry standard 16550A

device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, the UART is in M16C450 mode. Its FIFOs can be enabled and the UART can then enter M16550A mode. The UART adds further functionality beyond M16550A mode. Each of the extended functions can be selected individually under software control.

The UART provides more powerful enhancements than the industry-standard 16550:

- Hardware flow control. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.
- Output of an IR-compatible electrical pulse with a width 3/16 of that of a regular bit period.

Note: In order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The Enhanced Mode bit ensures that the UART is backward compatible with software that has been written for 16C450 and 16550A devices.

Figure 40 shows the block diagram of the UART device.

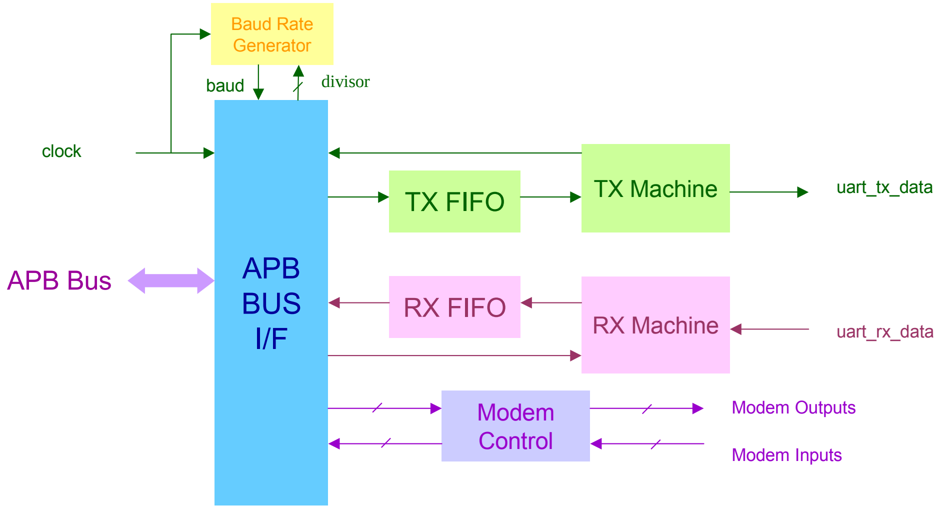


Figure 40 ***Block Diagram of UART***

2.29.2 Register Definitions

n = 1, 2, 3; for uart1, uart2 and uart3 respectively.

UARTn+0000h RX Buffer Register UARTn_RBR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RBR[7:0]**							
Type									RO							

RBR RX Buffer Register. Read-only register. The received data can be read by accessing this register. Modified when LCR[7] = 0.

UARTn+0000h TX Holding Register UARTn_THR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**THR[7:0]**							

 © 2009 MediaTek Inc.

Type									WO							

THR TX Holding Register. Write-only register. The data to be transmitted is written to this register, and then sent to the PC via serial communication.
Modified when LCR[7] = 0.

UARTn+0004h Interrupt Enable Register UARTn_IER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CTSI	RTSI	XOFFI	X	EDSSI	ELSI	ETBEI	ERBFI
Type									R/W							
Reset									0							

IER By storing a '1' to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt is disabled.
IER[3:0] are modified when LCR[7] = 0.
IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

CTSI Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.
Note: This interrupt is only enabled when hardware flow control is enabled.
- **0** Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.
- **1** Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line.

RTSI Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.
Note: This interrupt is only enabled when hardware flow control is enabled.
- **0** Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.
- **1** Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line.

XOFFI Masks an interrupt that is generated when an XOFF character is received.
Note: This interrupt is only enabled when software flow control is enabled.
- **0** Unmask an interrupt that is generated when an XOFF character is received.
- **1** Mask an interrupt that is generated when an XOFF character is received.

EDSSI When set ("1"), an interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.
- **0** No interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.
- **1** An interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

ELSI When set ("1"), an interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
- **0** No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
- **1** An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.

ETBEI When set ("1"), an interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO
have been reduced to its Trigger Level.
- **0** No interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level.

© 2009 MediaTek Inc.

1 An interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level

ERBFI When set ("1"), an interrupt is generated if the RX Buffer contains data.

0 No interrupt is generated if the RX Buffer contains data.

1 An interrupt is generated if the RX Buffer contains data.

UARTn+0008h Interrupt Identification Register UARTn_IIR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID4	ID3	ID2	ID1	ID0	NINT
Type									RO							
Reset									0	0	0	0	0	0	0	1

IIR Identify if there are pending interrupts; ID4 and ID3 are presented only when EFR[4] = 1.
The following table gives the IIR[5:0] codes associated with the possible interrupts:

IIR[5:0]	Priority Level	Interrupt	Source
000001	-	No interrupt pending	
000110	1	Line Status Interrupt	BI, FE, PE or OE set in LSR
000100	2	RX Data Received	RX Data received or RX Trigger Level reached.
001100	2	RX Data Timeout	Timeout on character in RX FIFO.
000010	3	TX Holding Register Empty	TX Holding Register empty or TX FIFO Trigger Level reached.
000000	4	Modem Status change	DDCD, TERI, DDSR or DCTS set in MSR
010000	5	Software Flow Control	XOFF Character received
100000	6	Hardware Flow Control	CTS or RTS Rising Edge

*Table **47 The IIR[5:0] codes associated with the possible interrupts***

Line Status Interrupt: A RX Line Status Interrupt (IIR[5:0`] == 000110b) is generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the Line Status Register.
RX Data Received Interrupt: A RX Received interrupt (IER[5:0] == 000100b) is generated if EFRBI (IER[0]) is set and either RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached. The interrupt is cleared by reading the RX Buffer Register or the RX FIFO (if enabled).
RX Data Timeout Interrupt:
When virtual FIFO mode is disabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO contains at least one character;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a CPU read from the RX FIFO.
The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading RX FIFO.
When virtual FIFO mode is enabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO is empty;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register.
RX Holding Register Empty Interrupt: A TX Holding Register Empty Interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX Holding Register or, if FIFOs are enabled, the TX FIFO becomes empty. The interrupt is cleared by writing to the TX Holding Register or TX FIFO if FIFO enabled.
Modem Status Change Interrupt: A Modem Status Change Interrupt (IIR[5:0] = 000000b) is generated if EDSSI (IER[3]) is set and either DDCD, TERI, DDSR or DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the Modem Status Register.
Software Flow Control Interrupt: A Software Flow Control Interrupt (IIR[5:0] = 010000b) is generated if Software Flow Control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the Interrupt Identification Register.
Hardware Flow Control Interrupt: A Hardware Flow Control Interrupt (IER[5:0] = 100000b) is generated if Hardware Flow Control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS Modem Control line. The interrupt is cleared by reading the Interrupt Identification Register.

UARTn+0008h FIFO Control Register UARTn_FCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1	RFTL0	TFTL1	TFTL0	DMA1	CLRT	CLRR	FIFOE
Type									WO							

FCR FCR is used to control the trigger levels of the FIFOs, or flush the FIFOs.
FCR[7:6] is modified when LCR != BFh
FCR[5:4] is modified when LCR != BFh & EFR[4] = 1
FCR[4:0] is modified when LCR != BFh

FCR[7:6] RX FIFO trigger threshold
0 1
1 6
2 12
3 **RXTRIG**

FCR[5:4] TX FIFO trigger threshold
0 1
1 4
2 8
3 14 (FIFOSIZE - 2)

DMA1 This bit determines the DMA mode, which the TXRDY and RXRDY pins support. TXRDY and RXRDY act to support single-byte transfers between the UART and memory (DMA mode 0) or multiple byte transfers (DMA mode1). Note that this bit has no effect unless the FIFOE bit is set as well

0 The device operates in DMA Mode 0.

1 The device operates in DMA Mode 1.

TXRDY – mode0: Goes active (low) when the TX FIFO or the TX Holding Register is empty. Becomes inactive when a byte is written to the Transmit channel.

TXRDY – mode1: Goes active (low) when there are no characters in the TX FIFO. Becomes inactive when the TX FIFO is full.

RXRDY – mode0: Becomes active (low) when at least one character is in the RX FIFO or the RX Buffer Register is full. Becomes inactive when there are no more characters in the RX FIFO or RX Buffer register.

RXRDY – mode1: Becomes active (low) when the RX FIFO Trigger Level is reached or an RX FIFO Character Timeout occurs. Goes inactive when the RX FIFO is empty.

CLRT Clear Transmit FIFO. This bit is self-clearing.

0 Leave TX FIFO intact.

1 Clear all the bytes in the TX FIFO.

CLRR Clear Receive FIFO. This bit is self-clearing.

0 Leave RX FIFO intact.

1 Clear all the bytes in the RX FIFO.

FIFOE FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect.

0 Disable both the RX and TX FIFOs.

1 Enable both the RX and TX FIFOs.

UARTn+000Ch Line Control Register UARTn_LCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLAB**	**SB**	**SP**	**EPS**	**PEN**	**STB**	**WLS1**	**WLS0**
Type									R/W							
Reset									0	0	0	0	0	0	0	0

LCR Line Control Register. Determines characteristics of serial communication signals.
Modified when LCR[7] = 0.

DLAB Divisor Latch Access Bit.

0 The RX and TX Registers are read/written at Address 0 and the IER register is read/written at Address 4.

1 The Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.

SB Set Break

0 No effect

1 SOUT signal is forced into the "0" state.

SP Stick Parity

0 No effect.

1 The Parity bit is forced into a defined state, depending on the states of EPS and PEN:
If EPS=1 & PEN=1, the Parity bit is set and checked = 0.
If EPS=0 & PEN=1, the Parity bit is set and checked = 1.

EPS Even Parity Select

0 When EPS=0, an odd number of ones is sent and checked.
1 When EPS=1, an even number of ones is sent and checked.

PEN Parity Enable
0 The Parity is neither transmitted nor checked.
1 The Parity is transmitted and checked.

STB Number of STOP bits
0 One STOP bit is always added.
1 Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.

WLS1, 0 Word Length Select.
0 5 bits
1 6 bits
2 7 bits
3 8 bits

UARTn+0010h Modem Control Register UARTn_MCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XOFF STATUS**	~~IR ENABLE~~	**X**	**LOOP**	**OUT2**	**OUT1**	**RTS**	**DTR**
Type									R/W							
Reset									0	0	0	0	0	0	0	0

MCR Modem Control Register. Control interface signals of the UART.
MCR[4:0] are modified when LCR[7] = 0,
MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.

XOFF Status This is a read-only bit.
0 When an XON character is received.
1 When an XOFF character is received.

LOOP Loop-back control bit.
0 No loop-back is enabled.
1 Loop-back mode is enabled.

OUT2 Controls the state of the output NOUT2, even in loop mode.
0 NOUT2=1.
1 NOUT2=0.

OUT1 Controls the state of the output NOUT1, even in loop mode.
0 NOUT1=1.
1 NOUT1=0.

RTS Controls the state of the output NRTS, even in loop mode.
0 NRTS=1.
1 NRTS=0.

DTR Control the state of the output NDTR, even in loop mode.
0 NDTR=1.
1 NDTR=0.

 © 2009 MediaTek Inc.

UARTn+0014h Line Status Register UARTn_LSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
Type									R/W							
Reset									0	1	1	0	0	0	0	0

LSR Line Status Register.
Modified when LCR[7] = 0.

FIFOERR RX FIFO Error Indicator.
- **0** No PE, FE, BI set in the RX FIFO.
- **1** Set to 1 when there is at least one PE, FE or BI in the RX FIFO.

TEMT TX Holding Register (or TX FIFO) and the TX Shift Register are empty.
- **0** Empty conditions below are not met.
- **1** If FIFOs are enabled, the bit is set whenever the TX FIFO and the TX Shift Register are empty. If FIFOs are disabled, the bit is set whenever TX Holding Register and TX Shift Register are empty.

THRE Indicates if there is room for TX Holding Register or TX FIFO is reduced to its Trigger Level.
- **0** **Reset whenever the contents of the TX FIFO are more than its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is not empty(FIFOs are disabled).**
- **1** Set whenever the contents of the TX FIFO are reduced to its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is empty and ready to accept new data (FIFOs are disabled).

BI Break Interrupt.
- **0** Reset by the CPU reading this register
- **1** If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits).
  If the FIFOs are enabled, this error is associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.

FE Framing Error.
- **0** Reset by the CPU reading this register
- **1** If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is the next to be read.

PE Parity Error
- **0** Reset by the CPU reading this register
- **1** If the FIFOs are disabled, this bit is set if the received data did not have a valid parity bit. If the FIFOs are enabled, the state of this bit is revealed when the referred byte is the next to be read.

OE Overrun Error.
- **0** Reset by the CPU reading this register.
- **1** If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the CPU before new data from the RX Shift Register overwrote the previous contents.

If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but not transferred to the FIFO.

DR Data Ready.

0 Cleared by the CPU reading the RX Buffer or by reading all the FIFO bytes.

1 Set by the RX Buffer becoming full or by a byte being transferred into the FIFO.

UARTn+0018h Modem Status Register UARTn_MSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DCD**	**RI**	**DSR**	**CTS**	**DDCD**	**TERI**	**DDSR**	**DCTS**
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									Input	Input	Input	Input	0	0	0	0

Note: After a reset, D4-D7 are inputs. A modem status interrupt can be cleared by writing '0' or set by writing '1' to this register. D0-D3 can be written to.

Modified when LCR[7] = 0.

MSR Modem Status Register

DCD Data Carry Detect.

When Loop = "0", this value is the complement of the NDCD input signal.

When Loop = "1", this value is equal to the OUT2 bit in the Modem Control Register.

RI Ring Indicator.

When Loop = "0", this value is the complement of the NRI input signal.

When Loop = "1", this value is equal to the OUT1 bit in the Modem Control Register.

DSR Data Set Ready

When Loop = "0", this value is the complement of the NDSR input signal.

When Loop = "1", this value is equal to the DTR bit in the Modem Control Register.

CTS Clear To Send.

When Loop = "0", this value is the complement of the NCTS input signal.

When Loop = "1", this value is equal to the RTS bit in the Modem Control Register.

DDCD Delta Data Carry Detect.

0 The state of DCD has not changed since the Modem Status Register was last read

1 Set if the state of DCD has changed since the Modem Status Register was last read.

TERI Trailing Edge Ring Indicator

0 The NRI input does not change since this register was last read.

1 Set if the NRI input changes from "0" to "1" since this register was last read.

DDSR Delta Data Set Ready

0 Cleared if the state of DSR has not changed since this register was last read.

1 Set if the state of DSR has changed since this register was last read.

DCTS Delta Clear To Send

0 Cleared if the state of CTS has not changed since this register was last read.

1 Set if the state of CTS has changed since this register was last read.

© 2009 MediaTek Inc.

UARTn+001Ch Scratch Register — UARTn_SCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SCR[7:0]**							
Type									R/W							

A general purpose read/write register. After reset, its value is un-defined.
Modified when LCR[7] = 0.

UARTn+0000h Divisor Latch (LS) — UARTn_DLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLL[7:0]**							
Type									R/W							
Reset									1							

UARTn+0004h Divisor Latch (MS) — UARTn_DLM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLM[7:0]**							
Type									R/W							
Reset									0							

Note: DLL & DLM can only be updated if DLAB is set (“1”).. Note too that division by 1 generates a BAUD signal that is constantly high.
Modified when LCR[7] = 1.
The table below shows the divisor needed to generate a given baud rate from CLK inputs of 6.5, 13, 26 MHz and 52 MHz. The effective clock enable generated is 16 x the required baud rate.

BAUD	6.5MHz	13MHz	26MHz	52MHz
110	3693	7386	14773	29545
300	1354	2708	5417	10833
1200	338	677	1354	2708
2400	169	338	677	1354
4800	85	169	339	677
9600	42	85	169	339
19200	21	42	85	169
38400	11	21	42	85
57600	7	14	28	56
115200	*	6	14	28

Table 48 Divisor needed to generate a given baud rate

UARTn+0008h Enhanced Feature Register — UARTn_EFR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**AUTO CTS**	**AUTO RTS**	**D5**	**ENABLE -E**	**SW FLOW CONT[3:0]**			
Type									R/W	R/W	R/W	R/W	R/W			
Reset									0	0	0	0	0			

*NOTE: Only when LCR=BF'h

Auto CTS Enables hardware transmission flow control

- **0** Disabled.
- **1** Enabled.

Auto RTS Enables hardware reception flow control

- **0** Disabled.
- **1** Enabled.

Enable-E Enable enhancement features.

- **0** Disabled.
- **1** Enabled.

CONT[3:0] Software flow control bits.

- **00xx** No TX Flow Control
- **10xx** Transmit XON1/XOFF1 as flow control bytes
- **01xx** Transmit XON2/XOFF2 as flow control bytes
- **11xx** Transmit XON1 & XON2 and XOFF1 & XOFF2 as flow control words
- **xx00** No RX Flow Control
- **xx10** Receive XON1/XOFF1 as flow control bytes
- **xx01** Receive XON2/XOFF2 as flow control bytes
- **xx11** Receive XON1 & XON2 and XOFF1 & XOFF2 as flow control words

UARTn+0010h XON1 — UARTn_XON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XON1[7:0]**							
Type									R/W							
Reset									0							

UARTn+0014h XON2 — UARTn_XON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XON2[7:0]**							
Type									R/W							
Reset									0							

UARTn+0018h XOFF1 — UARTn_XOFF1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XOFF1[7:0]**							
Type									R/W							
Reset									0							

UARTn+001Ch XOFF2 — UARTn_XOFF2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XOFF2[7:0]**							
Type									R/W							
Reset									0							

© 2009 MediaTek Inc.

*Note: XON1, XON2, XOFF1, XOFF2 are valid only when LCR=BFh.

UARTn+0020h AUTOBAUD_EN UARTn_AUTOBAUD_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																AUTO_EN
Type																R/W
Reset																0

AUTOBAUD_EN Auto-baud enable signal

0 Auto-baud function disable

1 Auto-baud function enable

UARTn+0024h HIGH SPEED UART UARTn_HIGHSPEED

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED [1:0]	
Type															R/W	
Reset															0	

SPEED UART sample counter base

0 based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL}

1 based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL}

2 based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL}

3 based on sampe_count * baud_pulse, baud_rate = system clock frequency / sampe_count

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 6.5M Hz based on different HIGHSPEED value.

BAUD	HIGHSPEED =	HIGHSPEED = 1	HIGHSPEED = 2
110	3693	7386	14773
300	1354	2708	7386
1200	338	677	2708
2400	169	338	677
4800	85	169	338
9600	42	85	169
19200	21	42	85
38400	11	21	42
57600	7	14	21
115200	*	7	14
230400	*	*	7
460800	*	*	*
921600	*	*	*

Table 49 ***Divisor needed to generate a given baud rate from 6.5MHz based on different HIGHSPEED value***

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 13 MHz based on different HIGHSPEED value.

BAUD	HIGHSPEED =	HIGHSPEED = 1	HIGHSPEED = 2
110	7386	14773	29545
300	2708	7386	14773
1200	677	2708	7386
2400	338	677	2708
4800	169	338	677
9600	85	169	338
19200	42	85	169
38400	21	42	85
57600	14	21	42
115200	7	14	21
230400	*	7	14
460800	*	*	7
921600	*	*	*

Table 50 ***Divisor needed to generate a given baud rate from 13MHz based on different HIGHSPEED value***

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 26 MHz based on different HIGHSPEED value.

BAUD	HIGHSPEED =	HIGHSPEED =	HIGHSPEED =
110	14773	29545	59091
300	5417	14773	29545
1200	1354	5417	14773
2400	677	1354	5417
4800	339	677	1354
9600	169	339	667
19200	85	169	339
38400	42	85	169
57600	28	42	85
115200	14	28	42
230400	7	14	28
460800	*	7	14

© 2009 MediaTek Inc.

921600	*	*	7

Table 51 ***Divisor needed to generate a given baud rate from 26 MHz based on different HIGHSPEED value***

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 52MHz based on different HIGHSPEED value.

BAUD	HIGHSPEED =	HIGHSPEED =	HIGHSPEED =
110	29545	59091	118182
300	10833	29545	59091
1200	2708	10833	29545
2400	1354	2708	10833
4800	677	1354	2708
9600	339	677	1354
19200	169	339	677
38400	85	169	339
57600	56	85	169
115200	28	56	85
230400	14	28	56
460800	7	14	28
921600	*	7	14

Table 52 Divisor needed to generate a given baud rate from 52 MHz based on different HIGHSPEED value

UARTn+0028h SAMPLE_COUNT UARTn_SAMPLE_COUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLECOUNT [7:0]**							
Type									R/W							
Reset									0							

When HIGHSPEED=3, the sample_count is the threshold value for UART sample counter (sample_num).
Count from 0 to sample_count.

UARTn+002Ch SAMPLE_POINT UARTn_SAMPLE_POINT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLEPOINT [7:0]**							
Type									R/W							
Reset									Ffh							

When HIGHSPEED=3, UART gets the input data when sample_count=sample_num.
e.g. system clock = 13MHz, 921600 = 13000000 / 14

sample_count = 14 and sample point = 6 (sample the central point to decrease the inaccuracy)

The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal.

UARTn+0030h AUTOBAUD_REG   UARTn_AUTOBAUD_REG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT[3:0]				BAUDRATE[3:0]			
Type									RO				RO			
Reset									0				0			

BAUD_RATE Autobaud baud rate

- **0** 115200
- 1 57600
- 2 38400
- 3 19200
- 4 9600
- 5 4800
- 6 2400
- 7 1200
- 8 300
- 9 110

BAUDSTAT Autobaud format

- **0** Autobaud is detecting
- 1 AT_7N1
- 2 AT_7O1
- 3 AT_7E1
- 4 AT_8N1
- 5 AT_8O1
- 6 AT_8E1
- 7 at_7N1
- 8 at_7E1
- 9 at_7O1
- 10 at_8N1
- 11 at_8E1
- 12 at_8O1
- 13 Autobaud detection fails

UARTn+0034h Rate Fix Address   UARTn_RATEFIX_AD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RESTRICT	FREQ_SEL	AUTOBAUD_RATE_FIX	RXTE_FIX
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

© 2009 MediaTek Inc.

RATE_FIX When you set "rate_fix"(34H[0]), you can transmit and receive data only if

1) the f13m_en is enable and the freq_sel (34H[2]) is set to 1, or
2) the f26m_en is enable and the freq_sel (34H[2]) is set to 0.

AUTOBAUD_RATE_FIX When you set "autobaud_rate_fix"(34H[1]), you can tx/rx the autobaud packet only if

1) the f13m_en is enable and the freq_sel (34H[2]) is set to 1, or
2) the f26m_en is enable and the freq_sel (34H[2]) is set to 0.

FREQ_SEL

0 Select f26m_en for rate_fix and autobaud_rate_fix

1 Select f13m_en for rate_fix and autobaud_rate_fix

RESTRICT The "restrict" (34H[3]) is used to set a more condition for the autobaud fsm starting point

UARTn+0038h AUTOBAUDSAMPLE UARTn_AUTOBAUDSAMPLE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										AUTOBAUDSAMPLE						
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										dh						

Since the system clock may change, autobaud sample duration should change as system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13.

UARTn+003Ch Guard time added register UARTn_GUARD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT[3:0]			
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

GUARD_CNT Guard interval count value. Guard interval = (1/(system clock / **div_step** / div)) * GUARD_CNT.

GUARD_EN Guard interval add enable signal.

0 No guard interval added.

1 Add guard interval after stop bit.

UARTn+0040h Escape character register UARTn_ESCAPE_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT[7:0]							
Type									WO							
Reset									FFh							

ESCAPE_DAT Escape character added before software flow control data and escape character, i.e. if tx data is xon (31h), with esc_en =1, uart transmits data as esc + CEh (~xon).

UARTn+0044h Escape enable register UARTn_ESCAPE_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

 © 2009 MediaTek Inc.

Name																ESC_EN
Type																R/W
Reset																0

ESC_EN Add escape character in transmitter and remove escape character in receiver by UART.

- **0** Do not deal with the escape character.
- **1** Add escape character in transmitter and remove escape character in receiver.

UARTn+0048h Sleep enable register UARTn_SLEEP_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SELLP_EN
Type																R/W
Reset																0

SLEEP_EN For sleep mode issue

- **0** Do not deal with sleep mode indicate signal
- **1** To activate hardware flow control or software control according to software initial setting when chip enters sleep mode. Releasing hardware flow when chip wakes up; but for software control, uart sends xon when awaken and when FIFO does not reach threshold level.

UARTn+004Ch Virtual FIFO enable register UARTn_VFIFO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																VFIFO_EN
Type																R/W
Reset																0

VFIFO_EN Virtual FIFO mechanism enable signal.

- **0** Disable VFIFO mode.
- **1** Enable VFIFO mode. When virtual mode is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt for DMA.

UARTn+0050h Rx Trigger Address UARTn_RXTRI_AD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG[3:0]			
Type													R/W			
Reset													0			

RXTRIG **When {rtm,rtl}=2'b11, The Rx FIFO threshold will be Rxtrig.**

UARTn+0054h Fractional Divider LSB Address UARTn_FRACDIV_L

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									R/W							
Reset									0	0	0	0	0	0	0	0

FRACDIV_L Add sampling count (+1) from state data7 to data0, in order to contribute fractional divisor.

UARTn+0058h Fractional Divider MSB Address UARTn_FRACDIV_M

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															R/W	
Reset															0	0

FRACDIV_M Add sampling count when in state stop to parity, in order to contribute fractional divisor.

FRACDIV_L / FRACDIV_L Add one sampling period to each symbol, in order to increase the baud rate accuracy.

2.30 USB 2.0 High-Speed Dual-Role Controller

2.30.1 General Description

The USB2.0 Controller can support 8 Tx and 8 Rx endpoints(excluding Endpoint 0). These endpoints can be individually configured in software to handle either Bulk transfers, Interrupt transfers or Isochronous transfers. There are 8 DMA channels and the embedded RAM size is 8Kbytes. The embedded RAM can be dynamically configured to each endpoint. When acting as the host for point-to-point communications, the controller maintains a frame counter and automatically schedules SOF, Isochronous, Interrupt and Bulk transfers.

Here is provided features.

- Operates either as the host/peripheral in point-to-point communications with another USB function or as a function controller for a USB peripheral
- Complies with the USB 2.0 standard for high-speed (480Mbps) functions and with the On-The-Go supplement to the USB 2.0 specification
- Certified for High-Speed OTG
- Supports point-to-point communications with one high-, full-, or low-speed device
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports Suspend and Resume signaling
- Supports High-Bandwidth Isochronous & Interrupt transfers
- UTMI+ Level 2 Transceiver Interface
- Synchronous RAM interface for FIFOs
- Support for DMA access to FIFOs
- Software connect/disconnect option
- Supports multi-layer operations on the AHB bus
- Performs all transaction scheduling in hardware

 © 2009 MediaTek Inc.

The USB2.0 Controller Block Diagram is as illustrated.

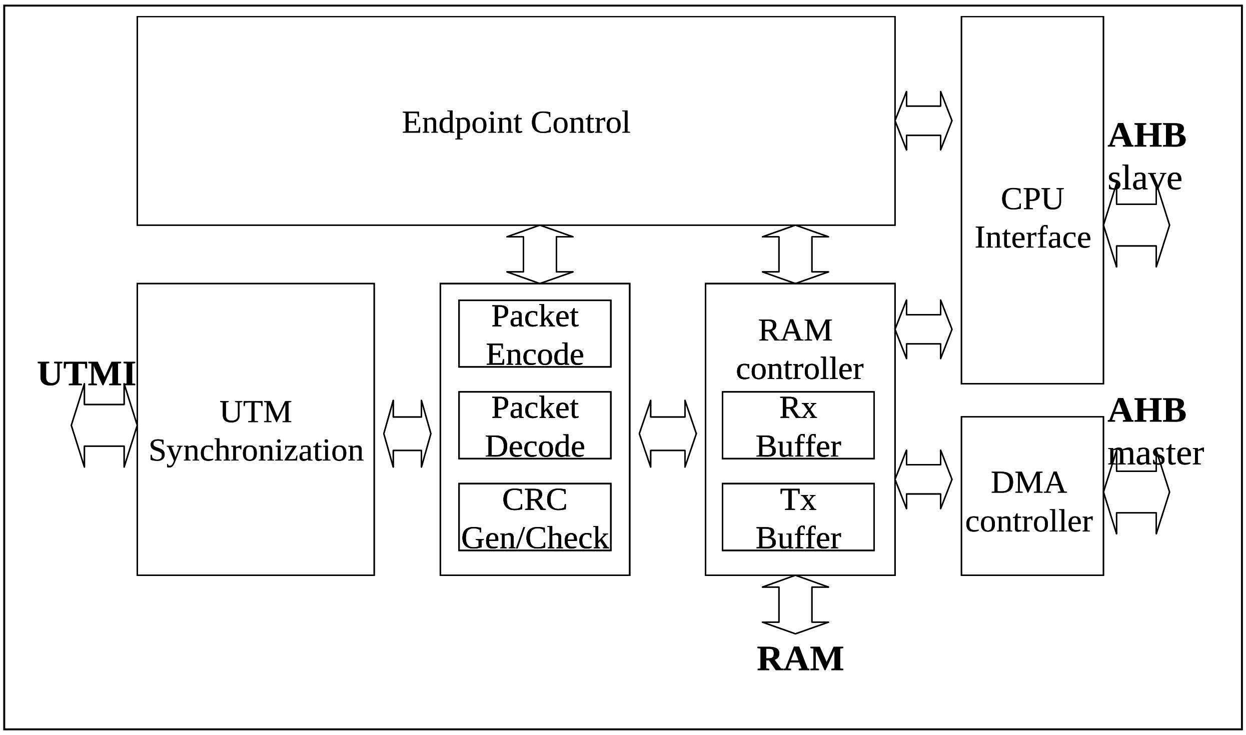


Figure 41 ***[Figure Caption]***

2.30.2 Register Definitions

REGISTER ADDRESS	REGISTER NAME	SYNONYM
USB + 0000h	Function Address Register	FADDR
USB + 0001h	Power Management Register	POWER
USB + 0002h	Tx Interrupt Status Register	INTRTX
USB + 0004h	Rx Interrupt Status Register	INTRRX
USB + 0006h	Tx Interrupt Enable Register	INTRTXE
USB + 0008h	Rx Interrupt Enable Register	INTRRXE
USB + 000Ah	Common USB interrupts Register	INTRUSB
USB + 000Bh	Common USB interrupts Enable Register	INTRUSBE
USB + 000Ch	Frame Number Register	FRAME
USB + 000Eh	Endpoint Selecting Index Register	INDEX
USB + 000Fh	Test Mode Enable Register	TESTMODE
USB + 0010h ~	It maps to CSR EP0 ~ EP5 depends on INDEX	Indexed CSR

© 2009 MediaTek Inc.

USB + 001Fh		
USB + 0020h	USB Endpoint 0 FIFO Register	FIFO0
USB + 0024h	USB Endpoint 1 FIFO Register	FIFO1
USB + 0028h	USB Endpoint 2 FIFO Register	FIFO2
USB + 002Ch	USB Endpoint 3 FIFO Register	FIFO3
USB + 0030h	USB Endpoint 4 FIFO Register	FIFO4
USB + 0034h	USB Endpoint 5 FIFO Register	FIFO5
USB + 0038h	USB Endpoint 6 FIFO Register	FIFO6
USB + 003Ch	USB Endpoint 7 FIFO Register	FIFO7
USB + 0040h	USB Endpoint 8 FIFO Register	FIFO8
USB + 0060h	OTG device control Register	DEVCTL
USB + 0061h	Power Up Counter Register	PWRUPCNT
USB + 0062h	Tx FIFO Size Register	TXFIFOSZ
USB + 0063h	Rx FIFO Size Register	RXFIFOSZ
USB + 0064h	Tx FIFO Address Register	TXFIFOADD
USB + 0066h	Rx FIFO Address Register	RXFIFOADD
USB + 006Ch	Hardware version Register	HWVERS
USB + 0070h	Software Reset Register	SWRST
USB + 0078h	Info. about number of Tx and Rx Register	EPINFO
USB + 0079h	Info. about the width of RAM and the number of DMA channel Register	RAMINFO
USB + 007Ah	Info. about delay to be applied Register	LINKINFO
USB + 007Bh	VBus Pulsing Charge Register	VPLEN
USB + 007Ch	Time buffer available on HS transactions Register	HS_EOF1
USB + 007Dh	Time buffer available on FS transactions Register	FS_EOF1
USB + 007Eh	Time buffer available on LS transactions Register	LS_EOF1
USB + 007Fh	RESET Information Register	RSTINFO
USB + 0102h	EP0 Control Status Register	CSR0
USB + 0108h	EP0 Received bytes Register	COUNT0
USB + 010Bh	NAK Limit Register	NAKLIMT0
USB + 010Fh	Core Configuration Register	CONFIGDATA
USB + 01n0h	TXMAP Register	TXMAP(n)
USB + 01n2h	Tx CSR Register	TXCSR(n)
USB + 01n4h	RXMAP Register	RXMAP(n)
USB + 01n6h	Rx CSR Register	RXCSR(n)
USB + 01n8h	Rx Count Register	RXCOUNT(n)
USB + 01nAh	TxType Register	TXTYPE(n)
USB + 01nBh	TxInterval Register	TXINTERVAL(n)

USB + 01nCh	RxType Register	RXTYPE(n)
USB + 01nDh	RxInterval Register	RXINTERVAL(n)
USB + 01nFh	Configured FIFO Size Register	FIFOSIZE(n)
	n stands for endpoint number. For example, endpoint 1's n = 1.	
USB + 0200h	DMA Interrupt Status Register	DMA_INTR
USB + 0204h	DMA Channel 1 Control Register	DMA_CNTL1
USB + 0208h	DMA Channel 1 ADDRESS Register	DMA_ADDR1
USB + 020Ch	DMA Channel 1 BYTE COUNT Register	DMA_COUNT1
USB + 0210h	DMA Channel 1 Limiter Register	DMA_LIMITER1
USB + 0214h	DMA Channel 2 Control Register	DMA_CNTL2
USB + 0218h	DMA Channel 2 ADDRESS Register	DMA_ADDR2
USB + 021Ch	DMA Channel 2 BYTE COUNT Register	DMA_COUNT2
USB + 0224h	DMA Channel 3 Control Register	DMA_CNTL3
USB + 0228h	DMA Channel 3 ADDRESS Register	DMA_ADDR3
USB + 022Ch	DMA Channel 3 BYTE COUNT Register	DMA_COUNT3
USB + 0234h	DMA Channel 4 Control Register	DMA_CNTL4
USB + 0238h	DMA Channel 4 ADDRESS Register	DMA_ADDR4
USB + 023Ch	DMA Channel 4 BYTE COUNT Register	DMA_COUNT4
USB + 023Ch	DMA Channel 5 Control Register	DMA_CNTL5
USB + 0240h	DMA Channel 5 ADDRESS Register	DMA_ADDR5
USB + 0244h	DMA Channel 5 BYTE COUNT Register	DMA_COUNT5
USB + 0248h	DMA Channel 6 Control Register	DMA_CNTL6
USB + 024Ch	DMA Channel 6 ADDRESS Register	DMA_ADDR6
USB + 0250h	DMA Channel 6 BYTE COUNT Register	DMA_COUNT6
USB + 0254h	DMA Channel 7 Control Register	DMA_CNTL7
USB + 0258h	DMA Channel 7 ADDRESS Register	DMA_ADDR7
USB + 025Ch	DMA Channel 7 BYTE COUNT Register	DMA_COUNT7
USB + 0260h	DMA Channel 8 Control Register	DMA_CNTL8
USB + 0264h	DMA Channel 8 ADDRESS Register	DMA_ADDR8
USB + 0268h	DMA Channel 8 BYTE COUNT Register	DMA_COUNT8
USB + 0284h	DMA Channel 1 PingPong Control Register	DMA_PP_CNTL1
USB + 0288h	DMA Channel 1 PingPong Address Register	DMA_PP_ADDR1
USB + 028Ch	DMA Channel 1 PingPong Count Register	DMA_PP_CNT1
USB + 0294h	DMA Channel 2 PingPong Control Register	DMA_PP_CNTL2
USB + 0298h	DMA Channel 2 PingPong Address Register	DMA_PP_ADDR2
USB + 029Ch	DMA Channel 2 PingPong Count Register	DMA_PP_CNT2
USB + 02A4h	DMA Channel 3 PingPong Control Register	DMA_PP_CNTL3

© 2009 MediaTek Inc.

USB + 02A8h	DMA Channel 3 PingPong Address Register	DMA_PP_ADDR3
USB + 02ACh	DMA Channel 3 PingPong Count Register	DMA_PP_CNT3
USB + 02B4h	DMA Channel 4 PingPong Control Register	DMA_PP_CNTL4
USB + 02B8h	DMA Channel 4 PingPong Address Register	DMA_PP_ADDR4
USB + 02BCh	DMA Channel 4 PingPong Count Register	DMA_PP_CNT4
USB + 02C0h	DMA Channel 5 PingPong Control Register	DMA_PP_CNTL5
USB + 02C4h	DMA Channel 5 PingPong Address Register	DMA_PP_ADDR5
USB + 02C8h	DMA Channel 5 PingPong Count Register	DMA_PP_CNT5
USB + 02CCh	DMA Channel 6 PingPong Control Register	DMA_PP_CNTL6
USB + 02D0h	DMA Channel 6 PingPong Address Register	DMA_PP_ADDR6
USB + 02D4h	DMA Channel 6 PingPong Count Register	DMA_PP_CNT6
USB + 02D8h	DMA Channel 7 PingPong Control Register	DMA_PP_CNTL7
USB + 02DCh	DMA Channel 7 PingPong Address Register	DMA_PP_ADDR7
USB + 02E0h	DMA Channel 7 PingPong Count Register	DMA_PP_CNT7
USB + 02E4h	DMA Channel 8 PingPong Control Register	DMA_PP_CNTL 8
USB + 02E8h	DMA Channel 8 PingPong Address Register	DMA_PP_ADDR 8
USB + 02ECh	DMA Channel 8 PingPong Count Register	DMA_PP_CNT8
USB + 0300h	EP1 RxPktCount Register	EP1RXPKTCOUNT
USB + 0302h	EP2 RxPktCount Register	EP2RXPKTCOUNT
USB + 0304h	EP3 RxPktCount Register	EP3RXPKTCOUNT
USB + 0308h	EP4 RxPktCount Register	EP4RXPKTCOUNT
USB + 030Ah	EP5 RxPktCount Register	EP5RXPKTCOUNT
USB + 030C h	EP6 RxPktCount Register	EP6RXPKTCOUNT
USB + 030Eh	EP7 RxPktCount Register	EP7RXPKTCOUNT
USB + 0310h	EP8 RxPktCount Register	EP8RXPKTCOUNT
USB + 0400h	DMA Channel 1 Real Count Register	DMA_REALCNT1
USB + 0404h	DMA Channel 1 PingPong Real Count Register	DMA_PP_REALCNT1
USB + 0408h	DMA Channel 1 Timer Register	DMA_Timer1
USB + 0410h	DMA Channel 2 Real Count Register	DMA_REALCNT2
USB + 0414h	DMA Channel 2 PingPong Real Count Register	DMA_PP_REALCNT2
USB + 0418h	DMA Channel 2 Timer Register	DMA_Timer2
USB + 0420h	DMA Channel 3 Real Count Register	DMA_REALCNT3
USB + 0424h	DMA Channel 3 PingPong Real Count Register	DMA_PP_REALCNT3
USB + 0428h	DMA Channel 3 Timer Register	DMA_Timer3
USB + 0430h	DMA Channel 4 Real Count Register	DMA_REALCNT4
USB + 0434h	DMA Channel 4 PingPong Real Count Register	DMA_PP_REALCNT4
USB + 0438h	DMA Channel 4 Timer Register	DMA_Timer4

© 2009 MediaTek Inc.

USB + 0440h	DMA Channel 5 Real Count Register	DMA_REALCNT5
USB + 0444h	DMA Channel 5 PingPong Real Count Register	DMA_PP_REALCNT5
USB + 0448h	DMA Channel 5 Timer Register	DMA_Timer5
USB + 0450h	DMA Channel 6 Real Count Register	DMA_REALCNT6
USB + 0454h	DMA Channel 6 PingPong Real Count Register	DMA_PP_REALCNT6
USB + 0458h	DMA Channel 6 Timer Register	DMA_Timer6
USB + 0460h	DMA Channel 7 Real Count Register	DMA_REALCNT7
USB + 0464h	DMA Channel 7 PingPong Real Count Register	DMA_PP_REALCNT7
USB + 0468h	DMA Channel 7 Timer Register	DMA_Timer7
USB + 0470h	DMA Channel 8 Real Count Register	DMA_REALCNT8
USB + 0474h	DMA Channel 8 PingPong Real Count Register	DMA_PP_REALCNT8
USB + 0478h	DMA Channel 8 Timer Register	DMA_Timer8
USB + 0600h	PHY Control Register 1	PHYCR1
USB + 0604h	PHY Control Register 2	PHYCR2
USB + 0608h	PHY Control Register 3	PHYCR3
USB + 060Ch	PHY Control Register 4	PHYCR4
USB + 0610h	PHY Control Register 5	PHYCR5
USB + 0614h	PHY UTMI Interface Register 1	PHYIR1
USB + 0618h	PHY UTMI Interface Register 2	PHYIR2
USB + 061ch	PHY UTMI Interface Register 3	PHYIR3

Table 53 ***[Table caption]***

USB COTROL REGISTER

USB+0000h Function Address Register FADDR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FUNCTION ADDRESS						
Type										R/W						
Reset										0						

FUNCTION ADDRESS FAddr is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction. When the USB2.0 controller is being used in Peripheral mode (DevCtl.bit2=0), this register should be written with the address received through a SET_ADDRESS command, which will then be used for decoding the function address in subsequent token packets. When the USB2.0 controller is being used in Host mode (DevCtl.bit2=1), this register should be set to the value sent in a SET_ADDRESS command during device enumeration as the address for the peripheral device.

Peripheral Mode

USB+0001h Power Management Register POWER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									ISOUPDATE	SOFTCONN	HSENAB	HSMODE	RESET	RESUME	SUSPENDMODE	ENABLE_SUSPENDM
Type									R/W	R/W	R/W	R	R	R/W	R	R/W
Reset									0	0	1	0	0	0	0	0

Host Mode

USB+0001h Power Management Register POWER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												HSENAB	HSMODE	RESUME	SUSPENDMODE	ENABLESUSPENDM
Type												R/W	R	R/W	Set	R/W
Reset												1	0	0	0	0

ENABLE_SUSPENDM Set by the CPU to enable the SUSPENDM output

SUSPENDMODE In Host mode, this bit is set by the CPU to enter Suspend mode. In Peripheral mode, this bit is set on entryo into Suspend mode. It is cleared when the CPU reads the interrupt register, or sets the Resume bit above.

RESUME Set by the CPU to generate Resume signaling when the function is in Suspend mode. The CPU should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling. In Host mode, this bit is also automatically set when Resume signaling from the target is detected while the USB2.0 controller is suspended.

RESET This bit is set when Reset signaling is present on the bus. Note: This bit is Read/Write from the CPU in Host Mode but Read-Only in Peripheral Mode.

HSMODE When set, this read-only bit indicates High-speed mode successfully negotiated during USB reset.
In Peripheral
Mode, becomes valid when USB reset completes (as indicated by USB reset interrupt). In Host Mode,
becomes valid when
Reset bit is cleared. Remains valid for the duration of the session.
Note: Allowance is made for Tiny-J signaling in determining the transfer speed to select.

HSENAB When set by the CPU, the USB2.0 controller will negotiate for High-speed mode when the device is reset by the hub. If not set, the device will only operate in Full-speed mode.

SOFTCONN If Soft Connect/Disconnect feature is enabled, then the USB D+/D- lines is enabled when this bit is set by the CPU and tri-stated when this bit is cleared by the CPU. Note: Only valid in Peripheral Mode.

ISOUPDATE When set by the CPU, the USB2.0 controller will wait for an SOF token from the time TxPktRdy
is set
before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will
be sent.Note:
Only valid in Peripheral Mode. Also, this bit only affects endpoints performing Isochronous transfers.

USB+0002h Tx Interrupt Status Register INTRTX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											EP5_TX	EP4_TX	EP3_TX	EP2_TX	EP1_TX	EP0
Type											R	R	R	R	R	R
Reset											0	0	0	0	0	0

EP0 Endpoint0 interrupt event

EP1_TX Tx Endpoint 1 interrupt event

EP2_TX Tx Endpoint 2 interrupt event

EP3_TX Tx Endpoint 3 interrupt event

EP4_TX Tx Endpoint 4 interrupt event

EP5_TX Tx Endpoint 4 interrupt event

USB+0004h Rx Interrupt Status Register INTRRX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													EP3_RX	EP2_RX	EP1_RX	
Type													R/W	R/W	R/W	
Reset													0	0	0	

INTRRX[15:0] Rx Interrupt Status register is “write 0 clear”

EP1_RX Rx Endpoint 1 interrupt event

EP2_RX Rx Endpoint 2 interrupt event

EP3_RX Rx Endpoint 3 interrupt event

USB+0006h Tx Interrupt Enable Register INTRTXE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											EP5_TXE	EP4_TXE	EP3_TXE	EP2_TXE	EP1_TXE	EP0_E
Type											R/W	R/W	R/W	R/W	R/W	R/W
Reset											1	1	1	1	1	1

EP0_E

- **0** Endpoint0 interrupt event disable
- **1** Endpoint0 interrupt event enable

EP1_TXE

- **0** Endpoint1 interrupt event disable
- **1** Endpoint1 interrupt event enable

EP2_TXE

- **0** Endpoint2 interrupt event disable
- **1** Endpoint2 interrupt event enable

EP3_TXE

- **0** Endpoint3 interrupt event disable
- **1** Endpoint3 interrupt event enable

EP4_TXE

© 2009 MediaTek Inc.

0 Endpoint4 interrupt event disable
1 Endpoint4 interrupt event enable

EP5_TXE

0 Endpoint5 interrupt event disable
1 Endpoint5 interrupt event enable

USB+0008h Rx Interrupt Enable Register INTRRXE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													EP3_RXE	EP2_RXE	EP1_RXE	
Type													R/W	R/W	R/W	
Reset													1	1	1	

EP1_RXE

0 Rx Endpoint1 interrupt event disable
1 Rx Endpoint1 interrupt event enable

EP2_RXE

0 Rx Endpoint1 interrupt event disable
1 Rx Endpoint1 interrupt event enable

EP3_RXE

0 Rx Endpoint1 interrupt event disable
1 Rx Endpoint1 interrupt event enable

USB+000Ah Common USB Interrupt Register INTRUSB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VBUSERROR	SESSREQ	DISCON	CONN	SOF	RESET/BABLE	RESUME	SUSPEND
Type									R	R	R	R	R	R	R	R
Reset									0	0	0	0	0	0	0	0

SUSPEND Set when Suspend signaling is detected on the bus. Only valid in Peripheral mode.
RESUME Set when Resume signaling is detected on the bus while the USB2.0 controller is in Suspend mode.
RESET Set in Peripheral mode when Reset signaling is detected on the bus.
BABBLE Set in Host mode when babble is detected. Note: Only active after first SOF has been sent.
SOF Set when a new frame starts.
CONN Set when a device connection is detected. Only valid in Host mode. Valid at all transaction speeds.
DISCON Set in Host mode when a device disconnect is detected. Set in Peripheral mode when a session ends. Valid at all
transaction speeds.
SESSREQ Set when Session Request signaling has been detected. Only valid when USB2.0 controller is ‘A’ device.
VBUSERROR Set when VBus drops below the VBus Valid threshold during a session. Only valid when USB2.0
controller is ‘A’ device.

USB+000Bh Common USB Interrupt Enable Register INTRUSBE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VBUSERROR_E	SESSREQ_E	DISCON_E	CONN_E	SOF_E	RESET/BABBLE_E	RESUM_E	SUSPEND_E
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0

SUSPEND_E Suspend interrupt enable.
RESUME_E Resume interrupt enable
RESET/BABBLE_E Reset/Babble interrupt enable
SOF_E SOF interrupt enable
CONN_E Conn interrupt enable
DISCON_E Discon interrupt enable
SESSREQ_E SessReq interrupt enable
VBUSERROR_E VBusError interrupt enable

USB+000Ch Frame Number Register FRAME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						FRAME NUMBER										
Type						R										
Reset						0										

FRAME_NUMBER Frame is a 11-bit read-only register that holds the last received frame number.

USB+000Eh Endpoint Selection Index Register INDEX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SELECTED ENDPOINT			
Type													R/W			
Reset													0			

SELECTED ENDPOINT Each Tx endpoint and each Rx endpoint have their own set of control/status registers located
between USB+100h – USB+1FFh. In addition one set of Tx control/status and one set of Rx control/status registers appear
at USB+010h – USB+01Fh. Index is a 4-bit register that determines which endpoint control/status registers are accessed.
Before accessing an endpoint's control/status registers at USB+010h – USB+01Fh, the endpoint number should be written
to the Index register to ensure that the correct control/status registers appear in the memory map.

USB+000Fh Test Mode Enable Register TESTMODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FORCE_HOST	FIFO_ACCESS	FORCE_FS	FORCE_HS	TEST_PACKET	TEST_K	TEST_J	TEST_SE0_NAK
Type									R/W	SET	R/W	R/W	R/W	R/W	R/W	R/W

Reset									0	0	0	0	0	0	0	0

TEST_SE0_NAK (HIGH-SPEED MODE) The CPU sets this bit to enter the Test_SE0_NAK test mode. In this mode, the
USB2.0 controller remains in High-speed mode but responds to any valid IN token with a NAK.

TEST_J (HIGH-SPEED MODE) The CPU sets this bit to enter the Test_J test mode. In this mode, the USB2.0 controller
transmits a continuous J on the bus.

TEST_K (HIGH-SPEED MODE) The CPU sets this bit to enter the Test_K test mode. In this mode, the USB2.0
controller transmits a continuous K on the bus.

TEST_PACKET (HIGH-SPEED MODE) The CPU sets this bit to enter the Test_Packet test mode. In this mode, the
USB2.0 controller repetitively transmits on the bus a 53-byte test packet, the form of which is defined in the Universal
Serial Bus Specification Revision 2.0, Section 7.1.20. Note: The test packet has a fixed format and must be loaded into the
Endpoint 0 FIFO before the test mode is entered.

FORCE_HS The CPU sets this bit either in conjunction with bit 7 above or to force the USB2.0 controller into High-speed
mode when it receives a USB reset.

FORCE_FS The CPU sets this bit either in conjunction with bit 7 above or to force the USB2.0 controller into Fullspeed
mode when it receives a USB reset.

FIFO_ACCESS The CPU sets this bit to transfer the packet in the Endpoint 0 Tx FIFO to the Endpoint 0 Rx FIFO. It is
cleared automatically.

FORCE_HOST The CPU sets this bit to instruct the core to enter Host mode when the Session bit is set, regardless of
whether it is connected to any peripheral. The state of the CID input, HostDisconnect and LineState signals are ignored.
The core will then remain in Host mode until the Session bit is cleared, even if a device is disconnected, and if the
Force_Host bit remains set, will re-enter Host mode the next time the Session bit is set.While in this mode, the status of the
HOSTDISCON signal from the PHY may be read from bit 7 of the ACTLR0.DevCtl register.
The operating speed is determined from the Force_HS and Force_FS bits as follows:

Force_HS	Force_FS	Operating Speed
0	0	Low Speed
0	1	Full Speed
1	0	High Speed
1	1	*Undefined*

Peripheral Mode

USB+0100h EP0 Control Status Register CSR0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**FLUSH FIFO**	**SERVI CESET UPED N**	**SERVI CEDR XPKTR DY**	**SEND STALL**	**SETUP END**	**DATA END**	**SEND STALL**	**TXPKT RDY**	**RXPK TRDY**
Type								SET	SET	SET	SET	R	SET	R/CLE AR	R/SET	R
Reset								0	0	0	0	0	0	0	0	0

RXPKTRDY This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the ServicedRxPktRdy bit.

TXPKTRDY The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).

SENTSTALL This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.

DATAEND The CPU sets this bit: When setting TxPktRdy for the last data packet. When clearing RxPktRdy after unloading the last data packet. When setting TxPktRdy for a zero length data packet.It is cleared automatically.

SETUPEND This bit will be set when a control transaction ends before the DataEnd bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the ServicedSetupEnd bit.

SENDSTALL The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically. Note: The FIFO should be flushed before SendStall is set.

SERVICERXPKTRDY The CPU writes a 1 to this bit to clear the RxPktRdy bit. It is cleared automatically.

SERVICESETUPEND The CPU writes a 1 to this bit to clear the SetupEnd bit. It is cleared automatically.

FLUSHFIFO The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. It is cleared automatically. The FIFO pointer is reset and the TxPktRdy/RxPktRdy bit (below) is cleared. Note: FlushFIFO should only be used when TxPktRdy/RxPktRdy is set. At other times, it may cause data to be corrupted.

Host Mode

USB+0102h EP0 Control Status Register CSR0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**DISPIN G**			**FLUSH FIFO**	**NAKTI MEOU T**	**STATU SPKT**	**REQP KT**	**ERRO R**	**SETUP PKT**	**RXSTA LL**	**TXPKT RDY**	**RXPK TRDY**
Type					R/W			SET	R/CLE AR	R/W	R/W	R/CLE AR	R/CLE AR	R/CLE AR	R/SET	R/CLE AR

Reset					0			0	0	0	0	0	0	0	0	0

RXPKTRDY This bit is set when a data packet has been received. An interrupt is generated (if enabled) when this bit is set.
The CPU should clear this bit when the packet has been read from the FIFO.

TXPKTRDY The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet
has been transmitted. An interrupt is also generated at this point (if enabled).

RXSTALL This bit is set when a STALL handshake is received. The CPU should clear this bit.

SETUPPKT The CPU sets this bit, at the same time as the TxPktRdy bit is set, to send a SETUP token instead of an OUT
token for the transaction. Note: Setting this bit also clears the DataToggle.

ERROR This bit will be set when three attempts have been made to perform a transaction with no response from the
peripheral. The CPU should clear this bit. An interrupt is generated when this bit is set.

REQPKT The CPU sets this bit to request an IN transaction. It is cleared when RxPktRdy is set.

STATUSPKT The CPU sets this bit at the same time as the TxPktRdy or ReqPkt bit is set, to perform a status stage
transaction. Setting this bit ensures that the data toggle is set to 1 so that a DATA1 packet is used for the Status Stage
transaction.

NAKTIMEOUT This bit will be set when Endpoint 0 is halted following the receipt of NAK responses for longer than the
time set by the NAKLimit0 register. The CPU should clear this bit to allow the endpoint to continue.

FLUSHFIFO The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The
FIFO pointer is reset and the TxPktRdy/RxPktRdy bit (below) is cleared. Note: FlushFIFO should only be used when
TxPktRdy/RxPktRdy is set. At other times, it may cause data to be corrupted.

DISPING The CPU writes a 1 to this bit to instruct the core not to issue PING tokens in data and status phases of a
high-speed Control transfer (for use with devices that do not respond to PING)

USB+0108h EP0 Received bytes Register COUNT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**EP0 RX COUNT**						
Type										R						
Reset										0						

EP0 RX COUNT0 Count0 is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RxPktRdy (IDXEPR0.CSR0.bit0) is set.

Host Mode

USB+010Bh NAK Limit Register NAKLIMT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												NAKLIMIT0				
Type												R/W				
Reset												0				

NAKLIMIT0 NAKLimit0 is a 5-bit register that sets the number of frames/microframes (High-Speed transfers) after which
Endpoint 0 should timeout on receiving a stream of NAK responses. (Equivalent settings for other endpoints can be made
through their TxInterval and RxInterval registers.). The number of frames/microframes selected is $2^{(m-1)}$ (where m is the
value set in the register, valid values 2 – 16). If the host receives NAK responses from the target for more frames than the
number represented by the Limit set in this register, the endpoint will be halted. Note: A value of 0 or 1 disables the NAK
timeout function.

USB+010Fh Core Configuration Register CONFIGDATA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MPRXE	MPTXE	BIGENDIAN	HBRXE	HBTXE	DYNFIFOSIZING	SOFTCONE	UTMIDATAWIDTH
Type									R	R	R	R	R	R	R	R
Reset									0	0	0	0	0	0	0	0

UTMIDATAWIDTH Indicates selected UTMI+ data width.

- **0** 8 bits
- **1** 16 bits

SOFTCONE When set to '1' indicates Soft Connect/Disconnect option selected.
DYNFIFOSIZEING When set to '1' indicates Dynamic FIFO Sizing option selected.
HBTXE When set to '1' indicates High-bandwidth Tx ISO Endpoint Support selected.
HBRXE When set to '1' indicates High-bandwidth Rx ISO Endpoint Support selected
BIGENDIAN When set to '1' indicates Big Endian ordering is selected.
MPTXE When set to '1', automatic splitting of bulk packets is selected.
MPRXE When set to '1', automatic amalgamation of bulk packets is selected.

USB+0110h TXMAP Register TXMAP

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M-1					MAXIMUM PAYLOAD TRANSACTION										
Type	R/W					R/W										
Reset	0					0										

TXMAXP M-1 Maxmum payload size for indexed TX endpoint
M-1 Packet multiplier m
TXMAXP MAXIMUM PAYLOD TRANSACTION REGISTER

The TxMaxP register defines the maximum amount of data that can be transferred through the selected Tx endpoint in asingle operation. There is a TxMaxP register for each Tx endpoint (except Endpoint 0).Bits 10:0 define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints

placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Fullspeed and High speed operations. Where the option of High-bandwidth Isochronous/Interrupt endpoints or of packet splitting on Bulk endpoints has been taken when the core is configured, the register includes either 2 or 5 further bits that define a multiplier m which is equal to one more than the value recorded.

In the case of Bulk endpoints with the packet splitting option enabled, the multiplier m can be up to 32 and defines the maximum number of 'USB' packets (i.e. packets for transmission over the USB) of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. (If the packet splitting option is not enabled, bit15–13 is not implemented and bit12–11(if included) is ignored.) Note: The data packet is required to be an exact multiple of the payload specified by bits 10:0, which is itself required to be either 8, 16, 32, 64 or (in the case of High Speed transfers) 512 bytes.

For Isochronous/Interrupt endpoints operating in High-Speed mode and with the High-bandwidth option enabled, m may only be either 2 or 3 (corresponding to bit 11 set or bit 12 set, respectively) and it specifies the maximum number of such transactions that can take place in a single microframe. If either bit 11 or bit 12 is non-zero, the USB2.0 controller will automatically split any data packet written to the FIFO into up to 2 or 3 'USB' packets, each containing the specified payload (or less). The maximum payload for each transaction is 1024 bytes, so this allows up to 3072 bytes to be transmitted in each microframe. (For Isochronous/Interrupt transfers in Full-speed mod, bits 11 and 12 are ignored.)

The value written to bits 10:0 (multiplied by m in the case of high-bandwidth Isochronous/Interrupt transfers) must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint (see USB

Specification Revision 2.0, Chapter 9). A mismatch could cause unexpected results.The total amount of data represented
by the value written to this register (specified payload × m) must not exceed the FIFO
size for the Tx endpoint, and should not exceed half the FIFO size if double-buffering is required.If this register is changed
after packets have been sent from the endpoint, the Tx endpoint FIFO should be completely flushed (using the FlushFIFO
bit in TxCSR) after writing the new value to this register.

Peripheral Mode

USB+0112h Tx CSR Register TXCSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTOSET	ISO	MODE	DMAREQEN	FRCDATATOG	DMAREQMODE	AUTOSETENSPKT		INCOMPTX	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	TXPKTRDY
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

TXPKTRDY The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet
has been transmitted. An interrupt is also generated at this point (if enabled). TxPktRdy is also automatically cleared (but no
interrupt is generated) prior to loading a second packet into a double-buffered FIFO.
FIFONOTEMPTY The USB sets this bit when there is at least 1 packet in the TxFIFO.
UNDERRUN The USB sets this bit if an IN token is received when the TxPktRdy bit not set. The CPU should clear this bit
(write 0 clear).
FLUSHFIFO The CPU writes a 1 to this bit to flush the latest packet from the endpoint TxFIFO. The FIFO pointer is reset,
the TxPktRdy bit is cleared and an interrupt is generated. May be set simultaneously with TxPktRdy to abort the packet that
is currently being loaded into the FIFO. Note: FlushFIFO should only be used when TxPktRdy is set. At other times, it may
cause data to be corrupted. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to ompletely
clear the FIFO.
SENDSTALL The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to
terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfer.
SENTSTALL This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TxPktRdy bit is cleared.
The CPU should clear this bit.
CLRDATATOG The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.

INCOMPTX When the endpoint is being used for high-bandwidth Isochronous/Interrupt transfers, this bit is set to indicate
where a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to
send all the parts. Note: In anything other than a high-bandwidth transfer, this bit will always return 0.

AUTOSETEN_SPKT If the CPU sets this bit, TxPktRdy will be automatically set when the short packet is loaded into the
TxFIFO completely. But, this function only works in Tx endpoint 1 and 2. Besides, Tx endpoint 1 has to use DMA channel
1 to move data and Tx endpoint 2 has to use DMA channel 2 to move data.

DMAREQMODE The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.Note:
This bit must not be cleared either before or in the same cycle as the DMAReqEn bit is cleared.

FRCDATATOG The CPU sets this bit to force the endpoint data toggle to switch and the data packet to be cleared from the
FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to
communicate rate feedback for Isochronous endpoints.

DMAREQEN The CPU sets this bit to enable the DMA request for the Tx endpoint.

MODE The CPU sets this bit to enable the endpoint direction as Tx, and clears the bit to enable it as Rx. Note: This bit only
has any effect where the same endpoint FIFO is used for both Tx and Rx transactions.

ISO The CPU sets this bit to enable the Tx endpoint for Isochronous transfers, and clears it to enable the Tx endpoint for
Bulk or Interrupt transfers. Note: This bit only has any effect in Peripheral mode. In Host mode, it always returns zero.

AUTOSET If the CPU sets this bit, TxPktRdy will be automatically set when data of the maximum packet size (value in
TxMaxP) is loaded into the TxFIFO. If a packet of less than the maximum packet size is loaded, then TxPktRdy will have
to be set manually if AutoSetEn_SPKT is not enabled.

Host Mode

USB+0112h Tx CSR Register TXCSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AUTOSET**		**MODE**	**DMAREQEN**	**FRCDATATOG**	**DMAREQMODE**			**NAKTIMEOUT/INCOMPTX**	**CLRDATAOG**	**RXSTALL**		**FLUSHFIFO**	**ERROR**	**FIFONOTEMPTY**	**TXPKTRDY**
Type	R/W		R/W	R/W	R/W	R/W			R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0		0	0	0	0			0	0	0		0	0	0	0

TXPKTRDY The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet

has been transmitted. An interrupt is also generated at this point (if enabled). TxPktRdy is also automatically cleared prior
to loading a second packet into a double-buffered FIFO.

FIFONOTEMPTY The USB sets this bit when there is at least 1 packet in the Tx FIFO.

ERROR The USB sets this bit when 3 attempts have been made to send a packet and no handshake packet has been
received. When the bit is set, an interrupt is generated, TxPktRdy is cleared and the FIFO is completely flushed. The CPU
should clear this bit. Valid only when the endpoint is operating in Bulk or Interrupt mode.

FLUSHFIFO The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer isreset,
the TxPktRdy bit (below) is cleared and an interrupt is generated. May be set simultaneously with TxPktRdy to abort the
packet that is currently being loaded into the FIFO. Note: FlushFIFO should only be used when TxPktRdy is set. At other
times, it may cause data to be corrupted. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set
twice to completely clear the FIFO.

RXSTALL This bit is set when a STALL handshake is received. When this bit is set, any DMA request that is in progress is
stopped, the FIFO is completely flushed and the TxPktRdy bit is cleared (see below). The CPU should clear this bit.

CLRDATATOG The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.

NAKTIMEOUT Bulk endpoints only: This bit will be set when the Tx endpoint is halted following the receipt of NAK
responses for longer than the time set as the NAK Limit by the TxInterval register. The CPU should clear this bit to allow
the endpoint to continue.

INCOMPTX High-bandwidth Interrupt endpoints only: This bit will be set if no response is received from the device to
which the packet is being sent.

DMAREQMODE The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.Note:
This bit must not be cleared either before or in the same cycle as the above DMAReqEnab bit is cleared.

FRCDATATOG The CPU sets this bit to force the endpoint data toggle to switch and the data packet to be cleared from the
FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to
communicate rate feedback for Isochronous endpoints.

DMAREQENAB The CPU sets this bit to enable the DMA request for the Tx endpoint.

MODE The CPU sets this bit to enable the endpoint direction as Tx, and clears it to enable the endpoint direction as Rx.

Note: This bit only has any effect where the same endpoint FIFO is used for both Tx and Rx transactions.
AUTOSET If the CPU sets this bit, TxPktRdy will be automatically set when a packet of the maximum packet size
(TxMaxP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then TxPktRdy will have
to be set manually. Note: Should not be set for either high-bandwidth Isochronous endpoints or high-bandwidth Interrupt
endpoints.

USB+0114h RXMAP Register RXMAP

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M-1					MAXIMUM PAYLOAD TRANSACTION										
Type	R/W					R/W										
Reset	0					0										

M-1 Maxmum payload size for indexed RX endpoint , M-1 Packet multiplier m

MAXIMUM PAYLOAD TRANSACTION REGISTER The RxMaxP register defines the maximum amount of data
that can be transferred through the selected Rx endpoint in a single operation. There is a RxMaxP register for each Rx
endpoint (except Endpoint 0).Bits 10:0 define (in bytes) the maximum payload transmitted in a single transaction. The
value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for Bulk,
Interrupt and Isochronous transfers in Full-speed and High-speed operations.
Where the option of High-bandwidth Isochronous/Interrupt endpoints or of combining Bulk packets has been taken when
the core is configured, the register includes either 2 or 5 further bits that define a multiplier m which is equal to one more
than the value recorded.
For Bulk endpoints with the packet combining option enabled, the multiplier m can be up to 32 and defines the number of
USB packets of the specified payload which are to be combined into a single data packet within the FIFO. (If the packet
splitting option is not enabled, bit15–bit13 is not implemented and bit12–bit11 (if included) is ignored.)
For Isochronous/Interrupt endpoints operating in High-Speed mode and with the High-bandwidth option enabled, m may
only be either 2 or 3 (corresponding to bit 11 set or bit 12 set, respectively) and it specifies the maximum number of such
transactions that can take place in a single microframe. If either bit 11 or bit 12 is non-zero, the USB2.0 controller will
automatically combine the separate USB packets received in any microframe into a single packet within the Rx FIFO. The

maximum payload for each transaction is 1024 bytes, so this allows up to 3072 bytes to be received in each microframe.
(For Isochronous/Interrupt transfers in Full-speed mode or if High-bandwidth is not enabled, bits 11 and 12 are ignored.)
The value written to bits 10:0 (multiplied by m in the case of high-bandwidth Isochronous/Interrupt transfers) must match
the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint (see USB
Specification Revision 2.0, Chapter 9). A mismatch could cause unexpected results.
The total amount of data represented by the value written to this register (specified payload × m) must not exceed the FIFO
size for the OUT endpoint, and should not exceed half the FIFO size if double-buffering is required.

Peripheral Mode

USB+0116h RX CSR Register RXCSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTOCLEAR	ISO	DMAREQEN	DISNYET/PIDERR	DMAREQMODE	AUTOCLRENSPKT	INCOMPRXINTREN	INCOMPRX	CLRDTATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERR	OVERRUN	FIFOFULL	RXPKTRDY
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXPKTRDY This bit is set when a data packet has been received (to RxFIFO). The CPU should clear this bit when the
packet has been unloaded from the RxFIFO. An interrupt is generated when the bit is set.
FIFOFULL This bit is set when no more packets can be loaded into the RxFIFO.
OVERRUN This bit is set if an OUT packet cannot be loaded into the RxFIFO. The CPU should clear this bit (write 0
clear). Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk Mode, it always returns zero. The
new incoming packet won't be written to RxFIFO. An interrupt is generated when the bit is set and OverRunIntrEn is set.
DATAERROR This bit is set when RxPktRdy is set if the data packet has a CRC or bit-stuff error. The CPU should write 0
to clear this bit. Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk Mode, it always returns
zero. An interrupt is generated when the bit is set and DataErrIntrEn is set.
FLUSHFIFO The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint RxFIFO. The RxFIFO
pointer is reset and the RxPktRdy bit is cleared. Note: FlushFIFO should only be used when RxPktRdy is set. At other
times, it may cause data to be corrupted. Also note that, if the RxFIFO is double buffered, FlushFIFO may need to be set
twice to completely clear the RxFIFO.

SENDSTALL The CPU writes a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminate the stall
condition. Note: This bit has no effect where the endpoint is being used for ISO transfers.

SENTSTALL This bit is set when a STALL handshake is transmitted. The CPU should clear this bit. An interrupt is
generated when the bit is set.

CLRDATATOG The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.

INCOMPRX This bit is set in a high-bandwidth Isochronous/Interrupt transfer if the packet in the RxFIFO is incomplete
because parts of the data were not received. It is cleared when RxPktRdy is cleared or write 0 to clear. Note: In anything
other than a high-bandwidth transfer, this bit will always return 0. An interrupt is generated when the bit is set and
IncompRxIntrEn is set.

INCOMPRXINTREN IncompRx and PidErr interrupt enable.

AUTOCLREN_SPKT The CPU write a 1 to this bit to enable short packets' RxPktRdy to be automatically cleared. Whe
this bit is turned on, AutoClear must also be turned on. If ISO and AutoClrEn_SPKT are both set, when short packets are
unloaded, RxPktRdy will be cleared automatically. But, these short packets must have no IncompRx, PidErr, DataErr or
OverRun status.

DMAREQMODE The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.
DMA Request Mode 1: Rx endpoint interrupt is generated only when DMA Request Mode 1 and received a short packet.
RxDMAReq is generated when receiving a Max-Packet-size packet.DMA Request Mode 0: No Rx endpoint interrupt.
RxDMAReq is generated when RxPktRdy is set.

DISNYET(BULK/INTERRUPT TRANSACTIONS) The CPU sets this bit to disable the sending of NYET handshakes.
When set, all successfully received Rx packets are ACK'd including at the point at which the RxFIFO becomes full. Note:
This bit only has any effect in High-speed mode, in which mode it should be set for all interrupt endpoint.

PIDERR(ISO TRANSACTIONS) This bit is set when there is a PID error in the received packet. It is cleared when
RxPktRdy is cleared or write 0 to clear. An interrupt is generated when the bit is set and IncompRxIntrEn is set.

DMAREQEN The CPU sets this bit to enable the DMA request for the Rx endpoint.

ISO The CPU sets this bit to enable the Rx endpoint for Isochronous transfers, and clears it to enable the Rx endpoint for
Bulk/Interrupt transfers.

AUTOCLEAR If the CPU sets this bit then the RxPktRdy bit will be automatically cleared when a packet of RxMaxP
bytes has been unloaded from the RxFIFO. When packets of less than the maximum packet size are unloaded, RxPktRdy
will have to be cleared manually if AutoClrEn_SPKT is not enabled.
Host Mode

USB+0116h Rx CSR Register RXCSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTO CLEAR	AUTO REQ	DMAR EQEN AB	PIDER ROR	DMAR EQMO DE			INCOM PRX	CLRD ATAT OG	RXSTA LL	REQP KT	FLUSH FIFO	DATA ERR/N AKTIM ER	ERRO R	FIFOF ULL	RXPK TRDY
Type	R/W	R/W	R/W	R	R/W			R/CLE AR	R/W	R/CLE AR	R/W	SET	R/CLE AR	R/CLE AR	R	R/CLE AR
Reset	0	0	0	0	0			0	0	0	0	0	0	0	0	0

RXPKTRDY This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been
unloaded from the Rx FIFO. An interrupt is generated when the bit is set.
FIFOFULL This bit is set when no more packets can be loaded into the Rx FIFO.
ERROR The USB sets this bit when 3 attempts have been made to receive a packet and no data packet has been received.
The CPU should clear this bit. An interrupt is generated when the bit is set. Note: This bit is only valid when the Rx
endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero.
NAKTIMEOUT In Bulk mode, this bit will be set when the Rx endpoint is halted following the receipt of NAK responses
for longer than the time set as the NAK Limit by the RxInterval register.
DATAERROR When operating in ISO mode, this bit is set when RxPktRdy is set if the data packet has a CRC or bit-stuff
error and cleared when RxPktRdy is cleared.
FLUSHFIFO The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO
pointer is reset and the RxPktRdy bit (below) is cleared. Note: FlushFIFO should only be used when RxPktRdy is set. At
other times, it may cause data to be corrupted. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set
twice to completely clear the FIFO.
REQPKT The CPU writes a 1 to this bit to request an IN transaction. It is cleared when RxPktRdy is set.
RXSTALL When a STALL handshake is received, this bit is set and an interrupt is generated. The CPU should clear this
bit.
CLRDATATOG The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.

 © 2009 MediaTek Inc.

INCOMPRX This bit will be set in a high-bandwidth Isochronous/Interrupt transfer if the packet received is incomplete. It
will be cleared when RxPktRdy is cleared. Note: If USB protocols are followed correctly, this bit should never be set. The
bit becoming set indicates a failure of the associated Peripheral device to behave correctly. (In anything other than a
high-bandwidth transfer, this bit will always return 0.)

DMAREQMODE The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.Note:
This bit should not be cleared in the same cycle as RxPktRdy is cleared

PIDERROR ISO Transactions Only: The core sets this bit to indicate a PID error in the received packet.
Bulk/Interrupt Transactions: The setting of this bit is ignored.

DMAREQENAB The CPU sets this bit to enable the DMA request for the Rx endpoint.

AUTOREQ If the CPU sets this bit, the ReqPkt bit will be automatically set when the RxPktRdy bit is cleared. Note: This
bit is automatically cleared when a short packet is received.

AUTOCLR f the CPU sets this bit then the RxPktRdy bit will be automatically cleared when a packet of RxMaxP bytes has
been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RxPktRdy will have
to be cleared manually. Note: Should not be set for highbandwidth Isochronous endpoints.

USB+0118h Rx Count Register RXCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**RXCOUNT**													
Type			R													
Reset			0													

RXCOUNT It is a 14-bit read-only register that holds the number of received data bytes in the packet in the RxFIFO. Note: The value returned changes as the FIFO is unloaded and is only valid while RxPktRdy(RxCSR.D0) is set.

Host Mode

USB+011Ah TxType Register TXTYPE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**TX PROTOCOL**		**TX TARGET EP NUMBER**			
Type											R/W		R/W			
Reset											0		0			

TX TARGET EP NUMBER (HOST MODE ONLY) The CPU should set this value to the endpoint number contained in the Tx endpoint descriptor returned to the USB2.0 Controller during device enumeration.

TX PROTOCOL (HOST MODE ONLY) The CPU should set this to select the required protocol for the Tx endpoint:

00 Illegal
01 Isochronous
10 Bulk
11 Interrupt

USB+011Bh TxInterval Register TXINTERVAL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TX POLLING INTERVAL/NAK LIMIT M							
Type									R/W							
Reset									0							

TxInterval Register TxInterval is an 8-bit register that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected Tx endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a TxInterval register for each configured Tx endpoint (except Endpoint 0).

TX POLLING INTERVAL / NAK LIMIT (M), (HOST MODE ONLY)
In each case the value that is set defines a number of frames/microframes (High Speed transfers), as follows:

Transfer Type	Speed	Valid values (*m*)	Interpretation
Interrupt	Low Speed or Full Speed	1 – 255	Polling interval is *m* frames.
	High Speed	1 – 16	Polling interval is $2^{(m-1)}$ microframes
Isochronous	Full Speed or High Speed	1 – 16	Polling interval is $2^{(m-1)}$ frames/microframes
Bulk	Full Speed or High Speed	2 – 16	NAK Limit is $2^{(m-1)}$ frames/microframes. *Note:* A value of 0 or 1 disables the NAK timeout function.

USB+011Ch RxType Register RXTYPE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											RX PROTOCOL		RX TARGET EP NUMBER			
Type											R/W		R/W			
Reset											0		0			

RX TARGET EP NUMBER (HOST MODE ONLY) The CPU should set this value to the endpoint number contained in the Tx endpoint descriptor returned to the USB2.0 Controller during device enumeration.

RX PROTOCOL (HOST MODE ONLY) The CPU should set this to select the required protocol for the Tx endpoint:

00 Illegal
01 Isochronous
10 Bulk
11 Interrupt

USB+011Dh RxInterval Register RXINTERVAL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RX POLLING INTERVAL/NAK LIMIT M**							
Type									R/W							
Reset									0							

RxInterval Register RxInterval is an 8-bit register that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected Rx endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a RxInterval register for each configured Rx endpoint (except Endpoint 0).

RX POLLING INTERVAL / NAK LIMIT (M), (HOST MODE ONLY)

In each case the value that is set defines a number of frames/microframes (High Speed transfers), as follows:

Transfer Type	Speed	Valid values (*m*)	Interpretation
Interrupt	Low Speed or Full Speed	1 – 255	Polling interval is *m* frames.
	High Speed	1 – 16	Polling interval is $2^{(m-1)}$ microframes
Isochronous	Full Speed or High Speed	1 – 16	Polling interval is $2^{(m-1)}$ frames/microframes
Bulk	Full Speed or High Speed	2 – 16	NAK Limit is $2^{(m-1)}$ frames/microframes. *Note:* A value of 0 or 1 disables the NAK timeout function.

Peripheral Mode

USB+011Fh Configured FIFO Size Register FIFOSIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RXFIFOSIZE**				**TXFIFOSIZE**			
Type									R				R			
Reset									0				0			

TXFIFOSIZE Indicate the TxFIFO size of 2n bytes, (ex: value 10 means 210 = 1024 bytes.)

RXFIFOSIZE Indicate the RxFIFO size of 2n bytes, (ex: value 10 means 210 = 1024 bytes.)

USB+0120h ~ USB+012Fh stands for Endpoint 2 Registers and their behaviors are the same as Endpoint 1.
USB+0130h ~ USB+013Fh stands for Endpoint 3 Registers and their behaviors are the same as Endpoint 1.
USB+0140h ~ USB+014Fh stands for Endpoint 4 Registers and their behaviors are the same as Endpoint 1.

USB+0020h USB Endpoint 0 FIFO Register FIFO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															
Reset	0															

 © 2009 MediaTek Inc.

USB+0024h USB Endpoint 1 FIFO Register FIFO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															
Reset	0															

USB+0028h USB Endpoint 2 FIFO Register FIFO2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															
Reset	0															

USB+002Ch USB Endpoint 3 FIFO Register FIFO3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															
Reset	0															

USB+0030h USB Endpoint 4 FIFO Register FIFO4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															
Reset	0															

USB+0034h USB Endpoint 5 FIFO Register FIFO5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FIFO DATA[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FIFO DATA[15:0]**															
Type	R/W															

© 2009 MediaTek Inc.

Reset	0

USB+0038h USB Endpoint 6 FIFO Register FIFO6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFO DATA[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO DATA[15:0]															
Type	R/W															
Reset	0															

USB+003Ch USB Endpoint 7 FIFO Register FIFO7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFO DATA[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO DATA[15:0]															
Type	R/W															
Reset	0															

USB+0040h USB Endpoint 8 FIFO Register FIFO8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFO DATA[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO DATA[15:0]															
Type	R/W															
Reset	0															

FIFOData 32-bits FIFO data access window

The Endpoint FIFO Registers provides 6 addresses for CPU access to the FIFOs for each endpoint. Writing to these

addresses loads data into the TxFIFO for the corresponding endpoint. Reading from these addresses unloads data from the

RxFIFO for the corresponding endpoint.

Note: (i) Transfers to and from FIFOs may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed

provided the data accessed is contiguous. However, all the transfers associated with one packet must be of the ame width

so that the data is consistently byte-, word- or double-word-aligned. The last transfer may however contain fewer bytes

than the previous transfers in order to complete an odd-byte or odd-word transfer.

 © 2009 MediaTek Inc.

(ii) Depending on the size of the FIFO and the expected maximum packet size, the FIFOs support either single-packet or double-packet buffering. However, burst writing of multiple packets is not supported as flags need to be set after each packet is written.

(iii) Following a STALL response or a Tx Strike Out error on Endpoint 0 – 4, the associated FIFO is completely flushed

USB+0060h OTG device control Register DEVCTL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									B_DEVICE	FSDEV	LSDEV			HOSTMODE	HOSTREQ	SESSION
Type									R/W	R/W	R/W			R/W	R/W	R/W
Reset									0	0	0			0	0	0

SESSION When operating as an 'A' device, this bit is set or cleared by the CPU to start or end a session. When operating as a 'B' device, this bit is set/cleared by the USB2.0 controller when a session starts/ends. It is also set by the CPU to initiate the Session Request Protocol. When the USB2.0 controller is in Suspend mode, the bit may be cleared by the CPU to perform a software disconnect. Note: Clearing this bit when the core is not suspended will result in undefined behavior.

HOSTREQ When set, the USB2.0 controller will initiate the Host Negotiation when Suspend mode is entered. It is cleared when Host Negotiation is completed. ('B' device only)

HOSTMODE This Read-only bit is set when the USB2.0 controller is acting as a Host.

LSDEV This Read-only bit is set when a low-speed device has been detected being connected to the port. Only valid in Host mode.

FSDEV This Read-only bit is set when a full-speed or high-speed device has been detected being connected to the port. (High-speed devices are distinguished from full-speed by checking for high-speed chirps when the device is reset.) Only valid in Host mode.

B_DEVICE This Read-only bit indicates whether the USB2.0 controller is operating as the 'A' device or the 'B' device.

- **0** 'A' device
- **1** 'B' device

Only valid while a session is in progress. Note: If the core is in Force_Host mode, this bit will indicate the state of the HOSTDISCON input signal from the PHY.

USB+0061h Power Up Counter Register PWRUPCNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**PWRUPCNT**			
Type													R/W			
Reset													4'hf			

PWRUPCNT[3:0] Power Up Counter Limit. Power Up Counter is used to count the K state duration during suspend and when it is timeout, the resume interrupt will be issued. The register should be configured according to AHB clock speed.

USB+0062h Tx FIFO Size Register TXFIFOSZ

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**TXDPB**	**TXSZ**			
Type												R/W	R/W			
Reset												0	0			

TXDPB Defines whether double-packet buffering supported for TxFIFO. When '1', double-packet buffering is supported. When '0', only single-packet buffering is supported.

TXSZ Maximum packet size to be allowed for (before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission). If TxDPB = 0, the FIFO will also be this size; if TxDPB = 1, the FIFO will be twice this size

TxSZ[3:0]				Packet Size (Bytes)
0	0	0	0	8
0	0	0	1	16
0	0	1	0	32
0	0	1	1	64
0	1	0	0	128
0	1	0	1	256
0	1	1	0	512
0	1	1	1	1024
1	0	0	0	2048 (Single-packet buffering only)
1	0	0	1	4096 (Single-packet buffering only)
1	1	1	1	3072 (Single-packet buffering only)

USB+0063h Rx FIFO Size Register RXFIFOSZ

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**RXDPB**	**RXSZ**			
Type												R/W	R/W			
Reset												0	0			

RXDPB Defines whether double-packet buffering supported for TxFIFO. When '1', double-packet buffering is supported.
When '0', only single-packet buffering is supported.

RXSZ Maximum packet size to be allowed for (before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission). If TxDPB = 0, the FIFO will also be this size; if TxDPB = 1, the FIFO will be twice this size

RxSZ[3:0]				Packet Size (Bytes)
0	0	0	0	8
0	0	0	1	16
0	0	1	0	32
0	0	1	1	64
0	1	0	0	128
0	1	0	1	256
0	1	1	0	512
0	1	1	1	1024
1	0	0	0	2048 (Single-packet buffering only)
1	0	0	1	4096 (Single-packet buffering only)
1	1	1	1	3072 (Single-packet buffering only)

USB+0064h Tx FIFO Address Register TXFIFOADD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**TXFIFOADD**													
Type			R/W													
Reset			0													

TXFIFOADD TxFIFOadd is a 13-bit register which controls the start address of the selected Rx endpoint FIFO.

TxFIFOadd[12:0]				Start Address
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
		…		…
1	F	F	F	FFF8

USB+0066h Rx FIFO Address Register RXFIFOADD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATAERRINTRE**	**OVERRUNITREN**		**RXFIFOADD**												
Type	R/W	R/W		R/W												
Reset	0	0		0												

RXFIFOADD RxFIFOadd is a 13-bit register which controls the start address of the selected Rx endpoint FIFO.

RxFIFOadd[12:0]				Start Address
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

© 2009 MediaTek Inc.

0	0	0	0	0
		...		...
1	F	F	F	FFF8

OVERRUNINTREN OverRun interrupt enable. The OverRun status bit is in RxCSR[2] and it should be write 0 to clear.

DATAERRINTREN DataErr interrupt enable. The DataErr status bit is in RxCSR[3] and it should be write 0 to clear.

USB+006Ch Version Register HWVERS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RC	XX					YYY									
Type	R	R					R									
Reset	0	0					0									

RC Set to '1' if RTL used from a Release Candidate rather than from a full release of the core.

XX Major Version Number (Range 0 – 31).

YYY Minor Version Number (Range 0 – 999).

HWVers register is a 16-bit read-only register that returns information about the version of the RTL from which the core
hardware was generated, in particular the RTL version number (vxx.yyy).

USB+0070h Software Reset Register SWRST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			OPSTATE									REDUCEDLY	UNDOSRPFIX	FRCVBUSVALID	SWRST	DISUSBRESET
Type			RO									R/W	R/W	R/W	R/W	R/W
Reset			0									0	0	0	0	0

DISUSBRESET The CPU sets this bit to Disable USBReset function. And, then the CPU can reset the hardware bySwRst.
USBReset will be asserted when doing High Speed Detection Handshake. (This bit will only be reset when hardware
reset.)

SWRST The CPU sets this bit to reset the endpoint and RAM interface hardware.

FRC_VBUSVALID The CPU sets this bit to force VBusVal = 1, VBusSess = 1 and VBusLo = 0.

UNDO_SRPFIX The CPU sets this bit to recover to the original circuit of USB2.0 IP about SRP.

REDUCEDLY The CPU can set this bit to reduce inter-pkt delay.

OPSTATE This register indicates the USB controller state information.

USB+0078h Info. about number of Tx and Rx Register EPINFO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RXENDPOINTS				TXENDPOINTS			
Type									R				R			
Reset									0				0			

TXENDPOINTS The number of Tx endpoints implemented in the design.

RXENDPOINTS The number of Rx endpoints implemented in the design.

This 8-bit read-only register allows read-back of the number of Tx and Rx endpoints included in the design.

USB+0079h Info. about the width of RAM and the number of DMA channel Register RAMINFO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DMACHANS				RAMBITS			
Type									R				R			
Reset									0				0			

RAMBITS The width of the RAM address bus – 1.

DMA CHANNELS The number of DMA channels implemented in the design.

This 8-bit read-only register provides information about the width of the RAM and the number of DMA channels.

USB+007Ah Info. about delay to be applied Register LINKINFO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									WTCON				WTID			
Type									R/W				R/W			
Reset									4'h5				4'hc			

WTID Sets the delay to be applied from IDPULLUP being asserted to IDDIG being considered valid in units of 4.369ms.

(The default setting corresponds to 52.43ms.)

WTCON Sets the wait to be applied to allow for the user's connect/disconnect filter in units of 533.3ns. (The default

setting corresponds to 2.667µs.) This 8-bit register allows some delays to be specified.

USB+007Bh Vbus Pulsing Charge Register VPLEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VPLEN							
Type									R/W							
Reset									8'h3C							

VPLEN Sets the duration of the VBus pulsing charge in units of 136.5 us. (The default setting corresponds to 8.19ms)

This 8-bit register sets the duration of the VBus pulsing charge.

USB+007Ch Time buffer available on HS transaction Register HS_EOF1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									HS_EOF1							
Type									R/W							
Reset									8'h80							

HS_EOF1 Sets for High-speed transactions the time before EOF to stop beginning new transactions, in units of 133.3ns. (The default setting corresponds to 17.07µs.)

USB+007Dh Time buffer available on FS transaction Register FS_EOF1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FS_EOF1							
Type									R/W							
Reset									8'h77							

FS_EOF1 Sets for Full-speed transactions the time before EOF to stop beginning new transactions, in units of 533.3ns. (The default setting corresponds to 63.46μs.)

USB+007Eh Time buffer available on LS transaction Register LS_EOF1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									LS_EOF1							
Type									R/W							
Reset									8'h72							

LS_EOF1 Sets for Low-speed transactions the time before EOF to stop beginning new transactions, in units of 1.067μs. (The default setting corresponds to 121.6μs.).

USB+007Fh Reset Information Register RSTINFO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									WTFSSE0				WTCHRP			
Type									R/W				R/W			
Reset									0				0			

WTCHRP Sets the delay to be applied from detecting Reset to sending chirp K (for Device only). The duration = 272.8 x WTChrp + 0.1 usec. (This register will only be reset when hardware reset.)

WTFSSE0 The field signifies the SE0 signal duration before issue the reset signal(for Device only). The duration = 272.8 x WTFSSE0 + 2.5 usec. (This register will only be reset when hardware reset.)

USB+0300h EP1 RxPktCount Register EP1RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EP1RXPKTCOUNT															
Type	R/W															
Reset	0															

EP1RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 1 size MaxP that are to be
transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when
AutoReq is not set.

USB+0302h EP2 RxPktCount Register EP2RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**EP2RXPKTCOUNT**
Type	R/W
Reset	0

EP2RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 2 size MaxP that are to be transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when AutoReq is not set.

USB+0304h EP3 RxPktCount Register EP3RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EP3RXPKTCOUNT**															
Type	R/W															
Reset	0															

EP3RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when AutoReq is not set.

USB+0308h EP4 RxPktCount Register EP4RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EP4RXPKTCOUNT**															
Type	R/W															
Reset	0															

EP4RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when AutoReq is not set.

USB+030Ah EP5 RxPktCount Register EP5RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EP5RXPKTCOUNT**															
Type	R/W															
Reset	0															

EP5RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when AutoReq is not set.

USB+030Ch EP6 RxPktCount Register EP6RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EP6RXPKTCOUNT															
Type	R/W															
Reset	0															

EP3RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be
transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when
AutoReq is not set.

USB+030Eh EP7 RxPktCount Register EP7RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EP3RXPKTCOUNT															
Type	R/W															
Reset	0															

EP7RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be
transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when
AutoReq is not set.

USB+0310h EP8 RxPktCount Register EP8RXPKTCOUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EP8RXPKTCOUNT															
Type	R/W															
Reset	0															

EP8RQPKTCOUNT (HOST MODE ONLY) Sets the number of packets of Rx Endpoint 3 size MaxP that are to be
transferred in a block transfer. Only used in Host mode when AutoReq is set. Has no effect in Peripheral mode or when
AutoReq is not set.
RqPktCount (Host Mode Only) For each Rx Endpoint 1 – 8, the USB2.0 controller provides a 16-bit RqPktCount register.
This read/write register is used in Host mode to specify the number of packets that are to be transferred in a block transfer
of one or more Bulk packets of length MaxP to Rx Endpoint n. The core uses the value recorded in this register to etermine

the number of requests to issue where the AutoReq option (included in the RxCSR register) has been set.Note: Multiple
packets combined into a single bulk packet within the FIFO count as one packet.

USB+0200h DMA Interrupt Status Register DMA_INTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PPB_FINISH8	PPA_FINISH8	PPB_FINISH7	PPA_FINISH7	PPB_FINISH6	PPA_FINISH6	PPB_FINISH5	PPA_FINISH5	PPB_FINISH4	PPA_FINISH4	PPB_FINISH3	PPA_FINISH3	PPB_FINISH2	PPA_FINISH2	PPB_FINISH1	PPA_FINISH1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMA_LIMITER												DMA_INTR			
Type	R												R/W			
Reset	0												0			

DMA_INTR Indicates pending DMA interrupts, one bit per DMA channel implemented. Bit 0 is used for DMA channel 1,
Bit 1 is used for DMA channel 2 etc. Write 0 clear.
PPA_FINISH1 Indicates dma channel 1 PingPongA finish status. Write 0 clear.
PPB_FINISH1 Indicates dma channel 1 PingPongB finish status. Write 0 clear.
PPA_FINISH2 Indicates dma channel 2 PingPongA finish status. Write 0 clear.
PPB_FINISH2 Indicates dma channel 2 PingPongB finish status. Write 0 clear.
PPA_FINISH3 Indicates dma channel 3 PingPongA finish status. Write 0 clear.
PPB_FINISH3 Indicates dma channel 3 PingPongB finish status. Write 0 clear.
PPA_FINISH4 Indicates dma channel 4 PingPongA finish status. Write 0 clear.
PPB_FINISH4 Indicates dma channel 4 PingPongB finish status. Write 0 clear.
PPA_FINISH5 Indicates dma channel 5 PingPongA finish status. Write 0 clear.
PPB_FINISH5 Indicates dma channel 5 PingPongB finish status. Write 0 clear.
PPA_FINISH6 Indicates dma channel 6 PingPongA finish status. Write 0 clear.
PPB_FINISH6 Indicates dma channel 6 PingPongB finish status. Write 0 clear.
PPA_FINISH7 Indicates dma channel 7 PingPongA finish status. Write 0 clear.
PPB_FINISH7 Indicates dma channel 7 PingPongB finish status. Write 0 clear.
PPA_FINISH8 Indicates dma channel 8 PingPongA finish status. Write 0 clear.
PPB_FINISH8 Indicates dma channel 8 PingPongB finish status. Write 0 clear.
DMA_LIMITER Please refer to USB+210 register .The DMA_LIMITER can be read in this address,too.

USB+0204h DMA Channel 1 Control Register DMA_CNTL1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			ENDMAMODE2	PP_RST	PP_EN	BURST MODE		BUSERR	ENDPNT				INTEN	DMAMODE	DMADIR	DMAEN
Type			R/W	R/W	R/W	R/W		R/W	R/W				R/W	R/W	R/W	R/W
Reset			0	0	0	0		0	0				0	0	0	0

DMA_EN Enable DMA. The bit will be cleared when the DMA transfer is completed.
DMA_DIR Direction. 0 : DMA Write(Rx endpoint), 1 : DMA Read(Tx endpoint).
DMA_MODE DMA Mode.

INT_EN Interrupt Enable.
EndPnt[3 :0] Endpoint number.
BUS_ERR Bus Error.
BURST_MODE Burst Mode.

- **00** Burst Mode 0 : Bursts of unspecified length.
- **01** Burst Mode 1 : INCR4 or unspecified length
- **10** Burst Mode 2 : INCR8, INCR4 or unspecified length.
- **11** Burst Mode 3 : INCR16, INCR8, INCR4 or unspecified length

PP_EN PingPong Buffer Enable.
PP_RST The CPU writes 1 to this bit to reset PingPong Buffer Sequence. The bit stands for current PingPong Buffer
Sequence when read.
ENDMAMODE2 Enable DMA mode 2 function. DMA mode 2: The short packets will be moved by DMA even the short
packets are not the last transfer of this DMA Count. The DMA mode 2 function can't be turned on when AutoReq = 1
DMA_CNTL2, DMA_CNTL3, DMA_CNTL4, DMA_CNTL5 and DMA_CNTL6 have the same modification.

USB+0208h DMA Channel 1 ADDRESS Register DMA_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMA_ADDR1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DMA_ADDR1[15:0]**															
Type	R/W															
Reset	0															

DMA_ADDR1 32bits DMA start address, updated (increase) by USB2.0 controller automatically while multiple packet
DMA (DMA Mode = 1) is used
DMA_ADDR2, DMA_ADDR3, DMA_ADDR4, DMA_ADDR5 and DMA_ADDR6 have the same modification.

USB+020Ch DMA Channel1 BYTE COUNT Register DMA_COUNT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMA COUNT1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DMA_COUNT1[15:0]**															
Type	R/W															
Reset	0															

DMA_COUNT1 32bits DMA transfer count with byte unit, updated (decrease) by USB2.0 controller automatically while
each packet is transferred. DMA_COUNT 2, DMA_COUNT3, DMA_COUNT4, DMA_COUNT5 and DMA_COUNT6

have the same modification.

USB+0214h ~ USB+021ch stands for DMA Channel 2 Registers and their behaviors are the same as DMA channel 1.

USB+0224h ~ USB+022ch stands for DMA Channel 3 Registers and their behaviors are the same as DMA channel 1.

USB+0234h ~ USB+023ch stands for DMA Channel 4 Registers and their behaviors are the same as DMA channel 1.

USB+0210h DMA Channel Limiter Register DMA_LIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMA_LIMITER															
Type	R/W															
Reset	0															

DMA_LIMITER This register is to suppress the Bus utilization of the DMA channel. The value is from 0 to 255. 0 means no limitation, and 255 means totally banned. The value between 0 and 255 means certain DMA can have permission to use AHB every (4 X n) AHB clock cycles.Note that it is not recommended to limit the Bus utilization of the DMA channels because this increases the latency of response to the masters, and the transfer rate decreases as well. Before using it, programmer must make sure that the bus masters have some protective mechanism to avoid entering the wrong states.

USB+0284h DMA Channel PingPong Control Register DMA_PP_CNTL1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DMAEN
Type																R/W
Reset																0

DMA_EN Enable DMA(PingPong Buffer DMA). The bit will be cleared when the DMA transfer is completed.

USB+0288h DMA Channel 1 PingPong Address Register DMA_PP_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMA_PP_ADDR1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	DMA_PP_ADDR1[15:0]
Type	R/W
Reset	0

DMA_PP_ADDR1[31:0] DMA(PingPong Buffer DMA) Channel 1 AHB Memory Address.

USB+028Ch DMA Channel 1 PingPong Count Register DMA_PP_CNT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMA_PP_CNT1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMA_PP_CNT1[15:0]															
Type	R/W															
Reset	0															

DMA_PP_CNT1[31:0] DMA(PingPong Buffer DMA) Channel 1 Byte Count.

USB+0294h ~ USB+029Ch stands for DMA Channel 2 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02A4h ~ USB+02ACh stands for DMA Channel 3 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02B4h ~ USB+02BCh stands for DMA Channel 4 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02C0h ~ USB+02C8h stands for DMA Channel 5 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02CCh ~ USB+02D4h stands for DMA Channel 6 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02D8h ~ USB+02E0h stands for DMA Channel 7 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+02E4h ~ USB+02ECh stands for DMA Channel 8 PingPong Registers and their behaviors are the same as DMA channel 1.

USB+0400h DMA Channel Real Count Register DMA_REALCNT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMA REALCNT[31:16]															
Type	R															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	DMA REALCNT[15:0]
Type	R
Reset	0

DMA_REALCNT[31:0] Indicate current transferred bytes of DMA channel 1.

USB+0404h DMA Channel 1 PingPong Real Count Register DMA_PP_REALCNT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMA PP REALCNT[31:16]															
Type	R															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMA PP REALCNT[15:0]															
Type	R															
Reset	0															

DMA_PP_REALCNT[31:0] Indicate current transferred bytes of DMA channel 1 PingPong.

USB+0408h DMA Channel 1 Timer Register DMA_TIMER1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TIMEOUTSTATUS	ENTIMER	REG TIMEOUT						
Type								R/W	R/W	R/W						
Reset								0	0	0						

ENTIMER Enable timer. When the timer is enabled and there is no this DMA transaction during the reg_timeout duration, then DMA interrupt will be issued. The timer will be reset whenever EnTimer = 0 or (DMA_EN = 0 and DAM_EN_PP = 0).

REG_TIMEOUT To config timeout duration. Timeout duration = 1280 * reg_timeout + 2.5 us.

TIMEOUT_STATUS Indicates the DMA channel has timeout situation. Write 0 clear.

USB+0410h ~ USB+0418h stands for DMA Channel 2 Registers and their behaviors are the same as DMA channel 1.

USB+0420h ~ USB+0428h stands for DMA Channel 3 Registers and their behaviors are the same as DMA channel 1.

USB+0430h ~ USB+0438h stands for DMA Channel 4 Registers and their behaviors are the same as DMA channel 1.

USB+0440h ~ USB+0448h stands for DMA Channel 5 Registers and their behaviors are the same as DMA channel 1.

USB+0450h ~ USB+0458h stands for DMA Channel 6 Registers and their behaviors are the same as DMA channel 1.

USB+0460h ~ USB+0468h stands for DMA Channel 7 Registers and their behaviors are the same as DMA channel 1.

USB+0470h ~ USB+0478h stands for DMA Channel 8 Registers and their behaviors are the same as DMA channel 1.

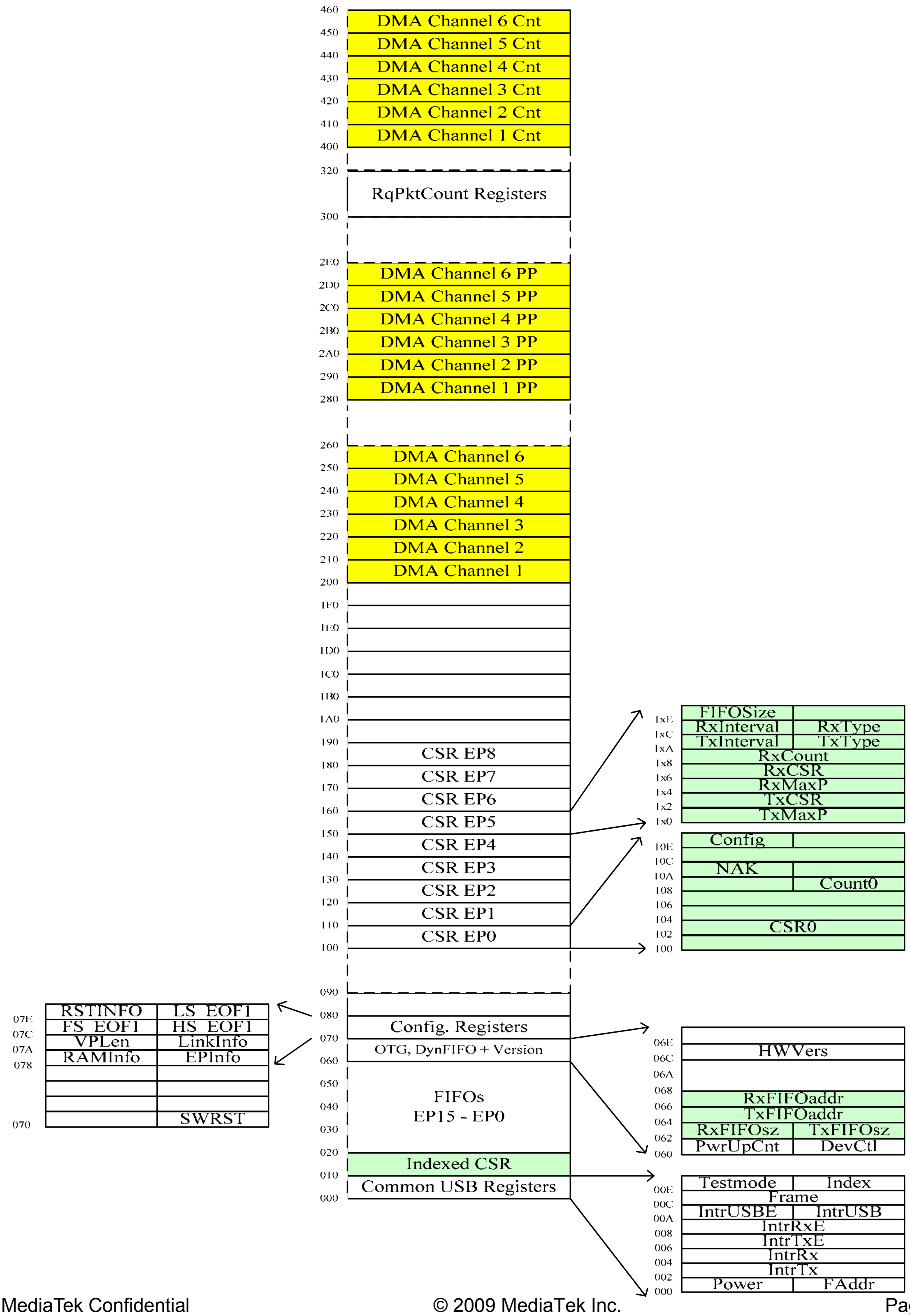

460
DMA Channel 6 Cnt
450
DMA Channel 5 Cnt
440
DMA Channel 4 Cnt
430
DMA Channel 3 Cnt
420
DMA Channel 2 Cnt
410
DMA Channel 1 Cnt
400
320
RqPktCount Registers
300
2E0
DMA Channel 6 PP
2D0
DMA Channel 5 PP
2C0
DMA Channel 4 PP
2B0
DMA Channel 3 PP
2A0
DMA Channel 2 PP
290
DMA Channel 1 PP
280
260
DMA Channel 6
250
DMA Channel 5
240
DMA Channel 4
230
DMA Channel 3
220
DMA Channel 2
210
DMA Channel 1
200
1F0
1E0
1D0
1C0
1B0
1A0
190
CSR EP8
180
CSR EP7
170
CSR EP6
160
CSR EP5
150
CSR EP4
140
CSR EP3
130
CSR EP2
120
CSR EP1
110
CSR EP0
100
090
080
Config. Registers
070
OTG, DynFIFO + Version
060
050
FIFOs
EP15 - EP0
040
030
020
Indexed CSR
010
Common USB Registers
000
1xE FIFOSize
1xC RxInterval RxType
1xA TxInterval TxType
1x8 RxCount
1x6 RxCSR
1x4 RxMaxP
1x2 TxCSR
1x0 TxMaxP
10E Config
10C
10A NAK
108 Count0
106
104
102 CSR0
100
07E RSTINFO LS_EOF1
07C FS_EOF1 HS_EOF1
07A VPLen LinkInfo
078 RAMInfo EPInfo
070 SWRST
06E
06C HWVers
06A
068 RxFIFOaddr
066 TxFIFOaddr
064 RxFIFOsz TxFIFOsz
062 PwrUpCnt DevCtl
060
00E Testmode Index
00C Frame
00A IntrUSBE IntrUSB
008 IntrRxE
006 IntrTxE
004 IntrRx
002 IntrTx
000 Power FAddr

Figure 42 ***[Figure Caption]***

USB+0600h PHY Control Register 1 PHYCR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HS_TX_IEN_MODE**	**HS_TX_SPSEL**	**HS_SQ_INIT_EN_DG[1:0]**		**HS_SQ_ENDG**	**HS_SQ_EN_MODE**	**HS_SQS**	**HS_RCV_EN_MODE**	**HS_ RCVB[3:0]**				**PLL_VCOG[1:0]**		**PLL_VCOB[1:0]**	
Type	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W				R/W		R/W	
Reset	1	0	2’b01		1	1	0	1	4’b0100				2’b01		0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PLL_CCP[3:0]**						**PLL_CLF**	**EN_LS_CMPSAT**	**PLL_EN**		**NEG_TRI_EN_B**	**USB20_TX_TST**	**BIDI_MODE**	**CDR_TST[1:0]**		**GATE_EN_B**
Type	R/W						R/W	R/W	R/W		R/W	R/W	R/W	R/W		R/W
Reset	4’b0110						0	0	0		0	0	0	0		0

GATED_EN_B High level clock gating enable

- **0** enable
- **1** disable

CDR_TST CDR function option ，CDR_TST[1]: phase accumulation option，

- **0** accumulation disable
- **1** accumulation

enable CDR_TST[0]: reference phase number option，0: 4 phases，1: 6 phases

BIDI_MODE UTMI data bus bi-directional mode

- **0** enable
- **1** disable

USB20_TX_TST TX macro test option, debug usage，

- **0** enable
- **1** disable

NEG_TRI_EN_B UTMI output signal aligned to negative edge for hold time issue

- **0** negative edge triggered output
- **1** positive edge triggered output

PLL_EN USB2.0 PHY PLL enable，

- **0** enable
- **1** disable

EN_LS_CMPSAT LS Tx mode DM RPU compensation enable when in client mode，

- **0** disable
- **1** enable

PLL_CLF PLL loop filter control，

- **0** disable
- **1** enable

PLL_CPP PLL CP bias current selection，charge pump current = 3.125uA * n，0001 : 3.125uA * 1，0010 : 3.125uA *

2…1111 : 3.125uA * 15 PLL_CCP should be set to 0x03 for better performance.

PLL_VCOB PLL VCO bias current selection，

- **00** 0uA
- **01** 25uA * 1.5
- **10** 25uA * 2.5
- **11** 25uA * 3.5

PLL_VCOG[0] PLL VCO gain selection ，

- **0** normal Kvco gain.
- **1** decrease Kvco gain

PLL_VCOG[1] This bit is used to gated internal clock source.

HS_RCVB HS RCV bias selection, HS RCV 1st stage bias current，

- **xxx0** 600u
- **xxx1** 675u

HS_RCV_EN_MODE HS RCV enable mode selection，

- **0** HS RCV enable by HS RCV
- **1** HS RCV always enabled

while USB operating in HS mode

HS_SQS HS SQ hystress mode Reserved

HS_SQ_EN_MODE HS SQ enable mode selection，

- **0** SQ enable by HS RCV
- **1** SQ always enabled while USB operating

HS_SQ_EN_DG HS SQ de-glitch time after HS RCV enabled，

- **0** SQ output de-glitch 4T 480M CLK
- **1** SQ output de-glitch 5T 480M CLK

HS_SQ_INIT_EN_DG HS SQ first time initializing de-glitch:gated by

- **00** 1T ref CLK
- **01** 1.5T ref CLK
- **10** 2T ref CLK
- **11** 2.5T ref CLK

HS_TX_SP_SEL HS TX LOAD sampling point selection，

- **0** sampling the HS TX data at rising edge
- **1** sampling the HS TX data at falling edge

HS_TX_I_EN_MODE HS TX I enable mode selection，

- **0** HS TX current always enabled while HS TERM enabled
- **1** HS_TX current enabled when HS TX module enabled

USB+0604h PHY Control Register 2 PHYCR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	FORCE_DRV_VBUS	FORCE_DM_PULLDOWN	FROCE_DP_PULLDOWN	HS_TERMC[4:0]					FIX_SONYBUG	FROCE_DATA_IN	FROCE_TXVALID	HS_TERM_SEL	HS_DISCTH[1:0]		HS_DISCB[1:0]	
Type	R/W	R/W	R/W	R/W						R/W	R/W	R/W	R/W		R/W	
Reset	0	0	0	5'b01000						0	0	0	0		0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SQTL[2:0]				HS_SQTH[2:0]			HS_SQD[3:0]				HS_SQB[3:0]			
Type		R/W				R/W			R/W				R/W			
Reset		3'b011				3'b100			4'b0010				4'b0010			

HS_SQB HS SQ bias selection，HS_SQB[0]: HS SQ 1st stage bias current，

xxx0 600uA
xxx1 675uA

HS_SQD HS SQ de-glitch control

xx00 16u (min current, max de-glitch)
xx01 32u
xx10 61u
xx11 122u (max current de-glitch)

HS_SQTH HS SQ threshold high selection

000 165mV
001 155mV
010 145mV
011 135mV
100 125mV
110 105mV
111 95mV

HS_SQTL HS SQ threshold low selection，Reserved

HS_DISCB HS_DISCP[0] : see HS_DISCN[1:0] ，HS_DISCP[1]: disconnect 1st stage bias current，

0 420uA
1 490uA

HS_DISCTH HS DISC threshold selection

000 615mV
001 605mV
010 595mV
011 585mV
100 575mV
101 565mV
110 555mV
111 545mV

HS_TERM_SEL HS TERM module selection (see HS_TERMC)，0: analog termination，1: digital termination

FORCE_TAVALID It is used to force PHY input signal TXVALID to a specific value. 1: enable. 0: disable.

FORCE_TAVALIDH It is used to force PHY input signal TXVALIDH to a specific value. 1: enable. 0: disable.

FORCE_DATAIN It is used to force PHY input signal DATAIN to a specific value. 1: enable. 0: disable.

FIX_SONYBUG When this bit is set , then TXFIFO write pointer will be reset when FlushFIFO. When this bit is not set, then TxFIFO write pointer won’t be reset when FlushFIFO.

HS_TERMC HS TERM impedence control code (see HS_TERM_SEL)，In analog termination mode

HS_TERM_SEL: 0

internal reference voltage:

x0000 480mV
x0001 470mV
x0010 460mV
x0011 450mV
x0100 440mV
x0101 430mV
x0110 420mV
x0111 410mV
x1000 400mV
x1001 390mV
x1010 380mV
x1011 370mV

the final output swing is about (the termnation voltage + 400mv)/2.

In digital termination mode HS_TERM_SEL: 1

00000 for max swing
…
11111 for min swing

FORCE_DP_PULLDOWN It is used to force PHY input signal DP_PULLDOWN to a specific value.

0 disable
1 enable

FORCE_DM_PULLDOWN It is used to force PHY input signal DM_PULLDOWN to a specific value.

0 disable
1 enable.

FORCE_DRV_VBUS It is used to force PHY input signal DRVVBUS to a specific value.

0 disable
1 enable

USB+0608h PHY Control Register 3 PHYCR3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		AIOI_SEL[2:0]			GHX_SEL[3:0]				CLKMODE[0]	XTAL_BIAS[2:0]			TEST_CTRL[3:0]			
Type		R/W			R/W				R/W	R/W			R/W			
Reset		0			0				0	0			0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PLL_DR[5:0]						FEN_HS_TX_I	FEN_FS_LS_RCV	FEN_FS_LS_TX	IREF_MODE_SEL	FEN_HS_RCV	IADJ[2:0]		

Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			6'b001010	0	0	0	0	0	3'b100

IADJ HS TX bias current selection

000 840mV(Rext + 300)
001 830mV(Rext + 300)
010 820mV(Rext + 300)
011 810mV(Rext + 300)
100 800mV(Rext + 300)
101 790mV(Rext + 300)
110 780mV(Rext + 300)
111 770mV(Rext + 300)

Rext = 5.1k (typ.)

FEN_HS_RCV Forced HS RCV and Squelch enable for test purpose

0 disable
1 enable

IREF_MODE_SEL HS SQ reference current mode selection ，0: HS SQ current always enabled while USB operating ，1: HS SQ current enabled while HS RX module enabled

FEN_FS_LS_TX Forced FS/LS output enable for test purpose

0 disable
1 enable

FEN_FS_LS_RCV Forced FS/LS RCV enable for test purpose，

0 disable
1 enable

FEN_HS_TX_I Forced HS TX current source enable for test purpose，

0 disable
1 enable

PLL_DR PLL div ratio 480/reference CLK=PLL_DR ex: 30MHz xtal case, PLL_DR = 480/30 = 16 = 0x10

TEST_CTRL Test mode control TEST_CTRL[3]:0: normal UTMI operation
1: 240MHz clock ouput, with while TEST_CTRL[0]=1 and FEN_HS_TX_I=11
TEST_CTRL[3]:

0 normal UTMI operation
1 240Mhz clock output ,with while TEST_CTRL[0]=1 and FEN_HS_TX_I=11

TEST_CTRL[2]:

0 no tx early
1 tx_early

TEST_CTRL[1]:

0 turn off 100K ohm register pull up DM signal
1 turn on 100K ohm register pull up DM signal

TEST_CTRL[0]:

0 normal UTMI operation

 © 2009 MediaTek Inc.

1 TX controlled by FEN_HS_TX_I/FEN_FS_LS_TX

XTAL_BIAS XTAL bias selection Reserved

CLKMODE External/Internal input CLK source selection

- **x00** internal clk source USB_INTA1_CK
- **x01** internal clk source USB_INTA2_CK
- **x10** internal clk source USB_INTD_CK

GHX_SEL GHX Digital output selection

- **xx00** none
- **xx01** HS DISC output for USB 2.0 RXDC test(set DOUT1_SEL)
- **xx10** HS SQ output for USB 2.0 RXDC test(set DOUT1_SEL)
- **xx11** HS RX output for USB 2.0 RXDC test(set DOUT1_SEL)

AIO1_SEL Analog IO1 selection for test (IO via XTALI pin)

- **0xxx** disable AIO1
- **100** external bgr input
- **101** monitor internal bgr voltage
- **110** monitor internal pll loop filter voltage
- **111** monitor internal hs termination control voltage

AIO ouput only valid for XTALI_GPIO_EN=1 & XTALI_GPIO_OE=0

USB+060Ch PHY Control Register 4 PHYCR4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**CLKMODE[1]**	**HS_DISCTH[2]**		**TX_FLUSH_EN**	**XTALI_GPIO_OE**	**XTALO_GPIO_ES**	**XTALO_GPIO_EN**	**XTALI_GPIO_I**	**XTALI_GPIO_OE**	**XTALI_GPIO_ES**	**XTALI_GPIO_EN**	**XTALI_GPIO_I**
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BGR_DIV[1:0]**			**BGR_SELPH**	**BGR_CHIP_EN**	**BGR_ISRC_EN**	**BGR_CLK_EN**	**BGR_BGR_EN**		**DOUT2_SEL[2:0]**				**DOUT1_SEL[2:0]**		
Type	R/W			R/W	R/W	R/W	R/W	R/W		R/W				R/W		
Reset	2'b01			0	1	0	0	0		3'b100				3'b100		

DOUT1_SEL Digital output 1 selection for test (output via VRT pin)

- **000** normal function
- **001** bgr ph1
- **010** bgr ph1s
- **011** bgr pheq
- **100** USB 2.0 RX DC test (see GHX_SEL)
- **101** USB 1.1 RXM
- **110** USB 1.1 RXP
- **111** USB 1.1 RXD

DOUT1 ouput only valid for XTALI_GPIO_EN=1 & XTALI_GPIO_OE=1

DOUT2_SEL Digital output 2 selection for test (output via VRT pin)

000 normal function
x01 bgr ph2
x10 bgr ph2s_
x11 bgr pho_
DOUT2 ouput only valid for XTALO_GPIO_EN=1 & XTALO_GPIO_OE=1

BGR_BGR_EN Force BGR enable 0: disable 1: enableBGRCLKEN

BGR_CLK_EN Force BGR chop clock enable 0: disable 1: enable

BGR_ISRC_EN Force BGR current source generator enable 0: disable 1: enable

BGR_CHP_EN BGR chop enable 0: disable 1: enable

BGR_DIV BGR chop clk rate

00 836k
11 836k/2
10 836k/4
11 836k/8

XTALI_GPIO_I It's used to control GPIO output data when XTALI pin GPIO function is enabled.
XTALI_GPIO_EN Enable XTALI pin GPIO function.
XTALI_GPIO_ES It's used to control GPIO output driving strength when XTALI pin GPIO function is enabled.
XTALI_GPIO_OE It's used to enable GPIO output function when XTALI pin GPIO function is enabled.
XTALO_GPIO_I It's used to control GPIO output data when XTALO pin GPIO function is enabled.
XTALO_GPIO_EN Enable XTALO pin GPIO function.
XTALO_GPIO_ES It's used to control GPIO output driving strength when XTALO pin GPIO function is enabled.
XTALO_GPIO_OE It's used to enable GPIO output function when XTALI pin GPIO function is enabled.
TX_FLUSH_EN When this bit is set , then TxFIFO write pointer will be reset when FlushFIFO. When this bit is not set, then TxFIFO write pointer won't be reset when FlushFIFO.

USB+0610h PHY Control Register 5 PHYCR5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DM_PULL_DOWN	DP_PULL_DOWN	XCVR_SELECT[1:0]		SUSEPNDM	TERM_SELECT	OP_MODE[1:0]		FROCE_IDPULLUP	UTMIMUXSEL	USB_MODE[1:0]		FROCE_XVER_SELECT	FROCE_SUSPENDM	FROCE_TERM_SELECT	FORCE_OP_MODE
Type	R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0		0	0	0		0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PROBE_SEL[7:0]								VBUS_CMP_EN	CLK_DIV_CNT[2:0]			CDR_FILT[3:0]			
Type	R/W								R/W	R/W			R/W			
Reset	0								1	0			4'b0010			

CDR_FILT CDR low pass filter selection, debug usage
CLK_DIVC_CNT The divide ratio of div_ck
PROBE_SEL Debug signal selection
FORCE_OPMODE It is used to force PHY input signal OPMODE to a specific value. 1: enable. 0: disable.

FORCE_TERM_SELECT It is used to force PHY input signal TERM_SELECT to a specific value. 1: enable. 0: disable.

FORCE_SUSPENDM It is used to force PHY input signal SUSPENDM to a specific value. 1: enable. 0: disable.

FORCE_XCVR_SELECT It is used to force PHY input signal XCVR_SELECT to a specific value. 1: enable. 0: disable.

USB_MODE Test mode selection (for testing)

- **00** normal operation
- **01** loop-back mode1 enable, the pseudo random number will be generated inside USB2.0 PHY macro and transmit onto USB bus. The data will be received by receiver and then be compared. The compared result is muxed on line_state[1] and should be always be 1.
- **10** loop-back mode2 enable, the packet is longer.

UTMI_MUXSEL It is used to force all PHY UTMI input signals to specific values

- **0** disable
- **1** enable

.

FORCE_IDPULLUP It is used to force PHY input signal IDPULLUP to a specific value.

- **0** disable
- **1** enable

OPMODE It is used to control PHY input signal OPMODE when force_opmode = 1 or utmi_muxsel = 1.

TERMSEL It is used to control PHY input signal TERMSEL when force_termsel = 1 or utmi_muxsel = 1.

SUSPENDM It is used to control PHY input signal SUSPENDM when force_suspendm = 1 or utmi_muxsel = 1.

XCVRSEL It is used to control PHY input signal XCVRSEL when force_xcvrsel = 1 or utmi_muxsel = 1.

DPPULLDOWN It is used to control PHY input signal DPPULLDOWN when force_dppulldown = 1 or utmi_muxsel = 1.

DMPULLDOWN It is used to control PHY input signal DMPULLDOWN when force_dmpulldown = 1 or utmi_muxsel = 1.

USB+0614h PHY UTMI Interface Register 1 PHYIR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LINESTATE[1:0]		HOSTDISC_ON	TXREADY	RXERROR	RXACTIVE	RXVALIDH	RXVALID	XDATA_OUT[15:8]							
Type	R		R	R	R	R	R	R	R							
Reset	0		0	0	0	0	0	0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XDATA_OUT[7:0]								XDATA_IN[3:0]				TX_VALIDH	TX_VALID	DRVVBUS	IDPULLUP
Type	R								R/W				R/W	R/W	R/W	R/W
Reset	0								0				0	0	0	0

IDPULLUP It is used to control PHY input signal IDPULLUP when force_idpullup = 1 or utmi_muxsel = 1.

DRVVBUS It is used to control PHY input signal DRVVBUS when force_drvvbus = 1 or utmi_muxsel = 1.

TX_VALID It is used to control PHY input signal TX_VALID when force_txvalid = 1 or utmi_muxsel = 1.

TX_VALIDH It is used to control PHY input signal TX_VALIDH when force_txvalidh = 1 or utmi_muxsel = 1.

XDATA_IN It is used to control PHY input signal XDATA_IN when force_datain = 1 or utmi_muxsel = 1.

XDATA_OUT It is used to control PHY input signal IDPULLUP when force_idpullup = 1 or utmi_muxsel = 1.

RXVALID It indicates the PHY output signal RXVALID status.

RXVALIDH It indicates the PHY output signal RXVALIDH status.

RXACTIVE It indicates the PHY output signal RXACTIVE status.

RXERROR It indicates the PHY output signal RXERROR status.

TXREADY It indicates the PHY output signal TXREADY status.

HOSTDISCON It indicates the PHY output signal HOSTDISCON status.

LINE_STATE It indicates the PHY output signal LINE_STATE status.

USB+0618h PHY UTMI Interface Register 2 PHYIR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FORCE_AUX_EN	FORCE_OTG_PROBE	FORCE_USB_CLKON	FORCE_BVALID	FORCE_IDDIG	FORCE_VBUSVALID	FORCE_SESSEND	FORCE_AVALID	USB_RESERVED[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
Reset	0	0	0	0	0	0	0	0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	USB_RESERVED[7:0]												IDDIG	VBUSVALID	SESSEND	AVALID
Type	R/W												R	R	R	R
Reset	0												0	0	0	0

FORCE_AVALID It is used to force PHY input signal AVALID to a specific value. 1: enable. 0: disable.
FORCE_SESSEND It is used to force PHY input signal SESSEND to a specific value. 1: enable. 0: disable.
FORCE_VBUSVALID It is used to force PHY input signal VBUSVALID to a specific value. 1: enable. 0: disable.
FORCE_IDDIG It is used to force PHY input signal IDIG to a specific value. 1: enable. 0: disable.
FORCE_BVALID It is used to force PHY input signal BVALID to a specific value. 1: enable. 0: disable.
FORCE_USB_CLKON It is used to force PHY input signal USB_CLKOFF to a specific value. 1: enable. 0: disable.
FORCE_OTG_PROBE It is used to force PHY input signal OTG_PROBE to a specific value. 1: enable. 0: disable.
FORCE_AUX_EN It is used to force PHY input signal AUX_EN to a specific value. 1: enable. 0: disable.
AVALID It indicates the OTG AVALID Comparator result.

0 avalid< 0.8v

1 avalid > 2v

SESSEND It indicates the OTG SESSEND Comparator result.

0 sessend< 0.2v

1 sessend > 0.8v

VBUSVALID It indicates the OTG VBUS Comparator result.

- **0** Vbus< 4.4v
- **1** Vbus > 4.75v

IDDIG It indicates the ID pin comparator result when IDPULLUP = 1.

- **0** A-plug is plugged in.
- **1** B-plug is plugged in

USB+061Ch PHY UTMI Interface Register 3 PHYIR3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUX_EN	OTG_PROBE	USB_CLKON	BVALID_W	IDDIG_W	VBUSVALID_W	SESSEND_W	AVALID_W
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1

AVALID_W It is used to control PHY input signal avalid when force_avalid = 1 or utmi_muxsel = 1

SESSEND_W It is used to control PHY input signal sessend when force_sessend = 1 or utmi_muxsel = 1.

VBUSVALID_W It is used to control PHY input signal vbusvalid when force_vbusvalid = 1 or utmi_muxsel = 1.

IDDIG_W It is used to control PHY input signal iddig when force_iddig = 1 or utmi_muxsel = 1.

BVALID_W It is used to control PHY input signal bvalid when force_bvalid = 1 or utmi_muxsel = 1.

USB_CLKON USB PHY input 48Mhz clock source signal enable 0: disable 1: enable

OTG_PROBE OTG debug signal enable 0: disable 1: enable

AUX_EN UART USB share pad function enable 0: disable 1: enable

2.30.3 Power on/off USB PHY and Controller Sequence

1 Power on sequence after plug-in.

1.1 Turn on Vusb(PHY 3.3v power) – the control register is in PMIC document.

1.2 Turn on USB AHB clock(52MHz) – the control register is in config document.

1.3 Turn on internal 48MHz PLL – the control register is in clock document.

1.4 Wait 50 usec. (PHY 3.3v power stable time)

1.5 It should set force_aux_en= 0 and force_usb_clkon = 1 to enable 48MHz PLL and switch to USB function. → reg[USB+061Bh] = 0x20.

1.6 Turn on Bandgap → reg[USB+060Dh] bit0 = 1. (bgr_bgr_en).

1.7 Wait 10 usec.

1.8 Release force suspendm. → reg[USB+0612h] bit2 = 0. (force_suspendm)

1.9 Wait 20 usec.

 © 2009 MediaTek Inc.

1.10 pll_en = 1. → reg[USB+0600h] bit7 = 1.

2 Power off sequence after plug-out.

2.1 Force suspendm = 0. → reg[USB+0613h] bit3 = 0.(suspendm) and reg[USB+0612h] bit2 = 1.(force_suspendm)

2.2 Wait 6 * 33.33ns

2.3 Pll_en = 0. → reg[USB+0600h] bit7 = 0

2.4 Turn off Bandgap → reg[USB+060Dh] bit0 = 0. (bgr_bgr_en).

2.5 Turn off Vusb(PHY 3.3v power) – the control register is in PMIC document

3 Initialzation Sequence

3.1 PLL_CCP[3:0]=0x3; (for better jitter performance).

3.2 TEST_CTRL[2] = 1; (for better jitter performance. Turn on tx earrlier).

 © 2009 MediaTek Inc.

3 Modem Micro-Controller Unit Subsystem

Figure 43 illustrates the block diagram of the Modem Micro-Controller Unit Subsystem in MT6516. The subsystem utilizes a main 32-bit ARM7EJ-S RISC processor, which plays the role of the main bus master controlling the whole subsystem. All processor transactions go to core cache first. The core cache controller accesses TCM (96KB memory dedicated to ARM7EJS core), cache memory, or bus according to the processor's request address. If the requested content is found in TCM or in cache, no bus transaction is required. If the core cache hit rate is high enough, bus traffic can be effectively reduced and processor core performance maximized. In addition to the benefits of reuse of memory contents, core cache also has a MPU (Memory Protection Unit), which allows cacheable and protection settings of predefined regions. The contents of core cache are only accessible to MCU.

The bus comprises of two-level system buses: Advanced High-Performance Bus (AHB) and Advanced Peripheral Bus (APB). All bus transactions originate from bus masters, while slaves can only respond to requests from bus masters. Before data transfer can be established, the bus master must ask for bus ownership, accomplished by request-grant handshaking protocol between masters and arbiters.

Two levels of bus hierarchy are designed to provide optimum usage for different performance requirements. Specifically, AHB Bus, the main system bus, is tailored toward high-speed requirements and provides 32-bit data path with multiplex scheme for bus interconnections. The APB Bus, on the other hand, is designed to reduce interface complexity for lower data transfer rate, and so it is isolated from high bandwidth AHB Bus by APB Bridge. APB Bus supports 16-bit addressing and 32-bit data path. APB Bus is also optimized for minimal power consumption by turning off the clock when there is no APB bus activity.

During operation, if the target slave is located on AHB Bus, the transaction is conducted directly on AHB Bus. However, if the target slave is a peripheral and is attached to the APB bus, then the transaction is conducted between AHB and APB bus through the use of APB Bridge.

The MT6516 Modem MCU subsystem supports only memory addressing method. Therefore all components are mapped onto the MCU 32-bit address space. A Memory Management Unit is employed to allow for a central decode scheme. The MMU generates appropriate selection signals for each memory-addressed module on the AHB Bus.

In order to off-load the processor core, a DMA Controller is designated to act as a master and share the bus resources on AHB Bus to perform fast data movement between modules. This controller provides five DMA channels.

The Interrupt Controller provides a software interface to manipulate interrupt events; it can handle up to 32 interrupt sources asserted at the same time. In general, the controller generates 2 levels of interrupt requests, FIQ and IRQ, to the processor.

For factory programming purposes, a Boot ROM module is also integrated. This module uses Internal Memory Controller to connect to AHB Bus.

MT6516 is fabricated using 65nm process. In deep-submicron process, leakage power dominates entire power consumption. Therefore, several power saving techniques are applied, such as MTCMOS (Multi-threshold CMOS). If the system is not working, power can be switched off to save standby leakage power. Modem MCU subsystem is a non-power-down subsystem, and other subsystem will be powered off if not using. Several submodules in other subsystem which cannot be powered down are moved into modem MCU subsystem, like shreg1, shreg2, tdma, afc, sleep_ctrl, etc, which locate in modem 2G subsystem originally.

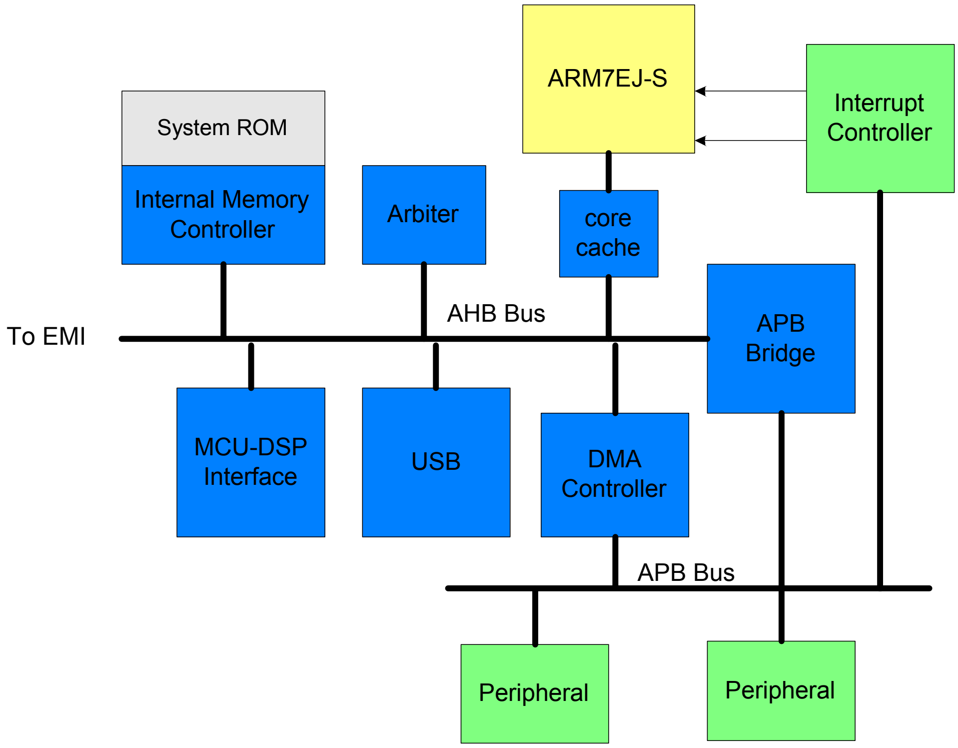


Figure 43 ***Block Diagram of the Modem Micro-Controller Unit Subsystem in MT6516***

3.1 Processor Core

3.1.1 General Description

The Modem Micro-Controller Unit Subsystem in MT6516 uses the 32-bit ARM7EJ-S RISC processor that is based on the Von Neumann architecture with a single 32-bit data bus that carries both instructions and data. The memory interface of ARM7EJ-S is totally compliant with the AMBA based bus system, which allows direct connection to the AHB Bus. In MT6516, ARM7EJ-S processor supports two operational frequencies : 104 & 52MHz, whereas in TK6516 ARM7 only operates in 104MHz.

3.2 Memory Management

3.2.1 General Description

The processor core of MT6516 supports only a memory addressing method for instruction fetch and data access. The core manages a 32-bit address space that has addressing capability of up to 4 GB. System ROM, Registers, MCU Peripherals and external components are all mapped onto such 32-bit address space, as depicted in **Figure 2-2**.

© 2009 MediaTek Inc.

BANK	Base Address	Description
BANK0	0000_0000h	EMI Bank 0 / Boot Code
BANK1	1000_0000h	EMI Bank 1
BANK2	2000_0000h	EMI Bank 2
BANK3	3000_0000h	EMI Bank 3
BANK4	4000_0000h	Reserved
	4800_0000h	System ROM
BANK5	5000_0000h	TCM
BANK6	6000_0000h	USB
	6100_0000h	Virtual FIFO
BANK7	7000_0000h	Reserved
BANK8	8000_0000h	APB Peripheral
BANK9	9000_0000h	Reserved
BANK10	A000_0000h	Share RAM IF 1
	A100_0000h	Share RAM IF 2
	A200_0000h	IDMA 1
	A300_0000h	IDMA 2
BANK11	B000_0000h	Reserved
BANK12	C000_0000h	Reserved
BANK13	D000_0000h	Reserved
BANK14	E000_0000h	Reserved
BANK15	F000_0000h	Reserved

Figure 44 ***The Memory Layout of MT6516***

The address space is organized into blocks of 256 MB each. The block number is uniquely selected by address line A31-A28 of the internal system bus.

3.2.1.1 External Access

To allow external access, the MT6516 outputs 27 bits (A26-A0) of address lines along with 4 selection signals that correspond to associated memory blocks. That is, MT6516 can support up to 4 MCU addressable external components. The data width of internal system bus is fixed at 32-bit wide, while the data width of the external components can be 8-, 16- or 32- bit.

Since devices are usually available with varied operating grades, adaptive configurations for different applications are needed. MT6516 provides software programmable registers to configure their wait-states to adapt to different operating conditions.

3.2.1.2 Memory Re-mapping Mechanism

To permit more flexible system configuration, a memory re-mapping mechanism is provided. The mechanism allows software program to swap BANK0 (ECS0#) and BANK1 (ECS1#) dynamically. Whenever the bit value of RM0 in register EMI_GENA is changed, these two banks are swapped accordingly. Furthermore, it allows system to boot from System ROM as detailed in 2.2.1.3 Boot Sequence.

3.2.1.3 Boot Sequence

Since the ARM7EJ-S core always starts to fetch instructions from the lowest memory address at 00000000h after system has been reset, the system is designed to have a dynamic mapping architecture capable of associating Boot Code, external Flash or external SRAM with the memory block 0000_0000h – 0fff_ffffh.
Both Modem side ARM7EJ-S and AP side ARM926EJ-S start to fetch instruction at 0x00000000h and execute the same program by default. If two cores would like to execute separate codes respectively, register EMI_GENE bit 0-15 should be configured to specify the offset address value for modem side MCUSYS.
By default, the Boot Code is mapped onto 0000_0000h – 0fff_ffffh after a system reset. In this special boot mode, External Memory Controller does not access external memory; instead, the EMI Controller send predefined Boot Code back to the ARM7EJ-S core, which instructs the processor to execute the program in System ROM. This configuration can be changed by programming bit value of RM1 in register EMI_GENA directly.
MT6516 system provides one boot up scheme:

- Start up system of running codes from Boot Code for factory programming or NAND flash boot.

3.2.1.3.1 Boot Code

The Boot Code is placed together with Memory Re-Mapping Mechanism in External Memory Controller, and comprises of just two words of instructions as shown below. A jump instruction leads the processor to run the code starting at address 4800_0000h where the System ROM is placed.

ADDRESS	BINARY CODE	ASSEMBLY	
00000000h	E51FF004h	LDR	PC, 0x4
00000004h	48000000h	(DATA)	

3.2.1.3.2 Factory Programming

The configuration for factory programming is shown in **Figure 2-3**. Usually the Factory Programming Host connects with MT6516 via the UART interface. The download speed can be up to 921K bps while MCU is running at 26MHz.
After the system has reset, the Boot Code guides the processor to run the Factory Programming software placed in System ROM. Then, MT6516 starts and polls the UART1 port until valid information is detected. The first information received on the UART1 is used to configure the chip for factory programming. The Flash downloader program is then transferred into System RAM or external SRAM.
Further information is detailed in the MT6516 Software Programming Specification.

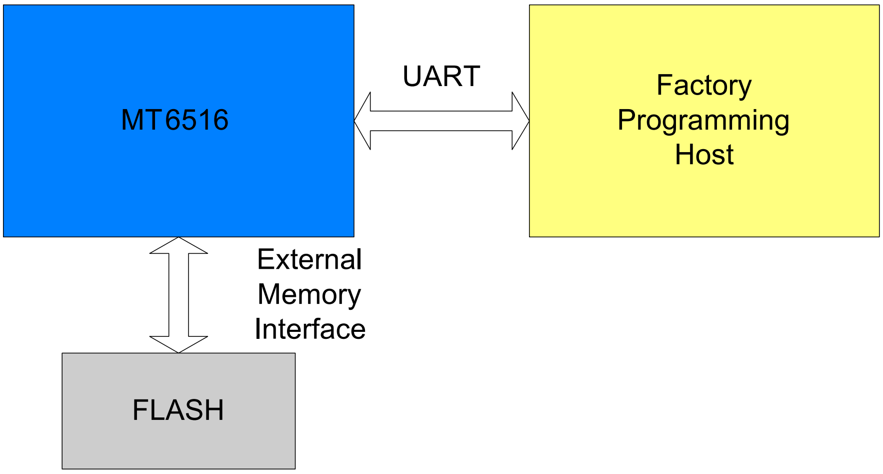


Figure 45 ***System configuration required for factory programming***

3.2.1.3.3 NAND Flash Booting

If MT6516 cannot receive data from UART1 for a certain amount of time, the program in System ROM checks if any valid boot loader exists in NAND flash. If found, the boot loader code is copied from NAND flash to RAM (external) and executed to start the real application software. If no valid boot loader can be found in NAND flash, MT6516 starts executing code in EMI bank0 memory. The whole boot sequence is shown in the following figure.

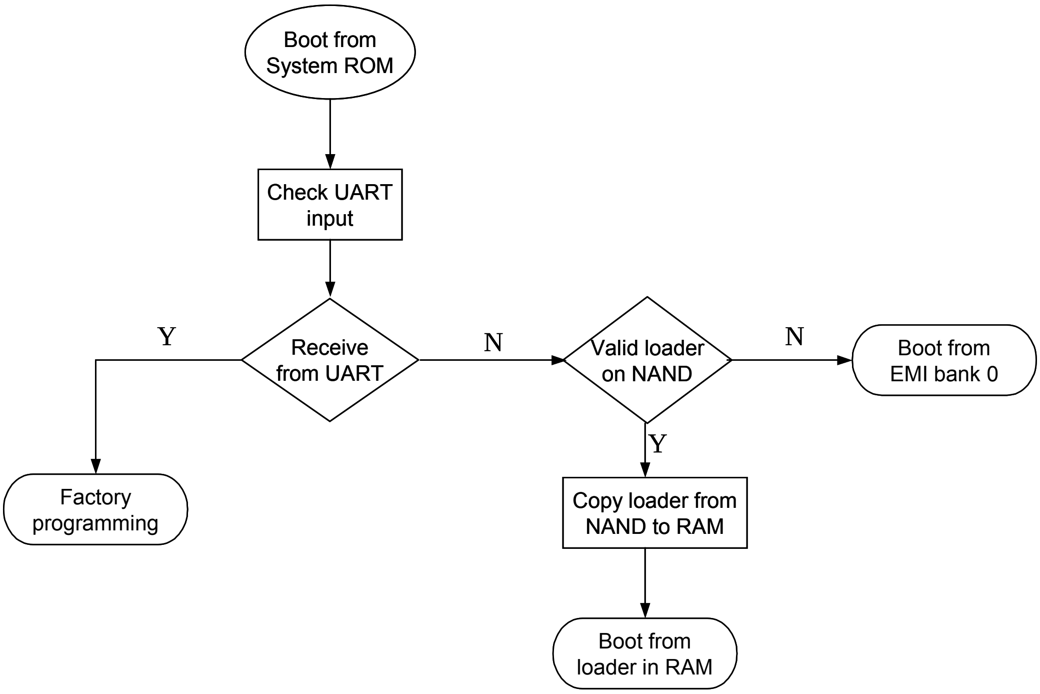


Figure 46 ***Boot sequence***

© 2009 MediaTek Inc.

3.2.1.4 Little Endian Mode

The MT6516 system always treats 32-bit words of memory in Little Endian format. In Little Endian mode, the lowest numbered byte in a word is stored in the least significant position, and the highest numbered byte in the most significant position. Byte 0 of the memory system is therefore connected to data lines 7 through 0.

3.3 Bus System

3.3.1 General Description

Two levels of bus hierarchy are employed in the Modem Micro-Controller Unit Subsystem of MT6516. As depicted in **Figure 43**, AHB Bus and APB Bus serve as system backbone and peripheral buses, while an APB bridge connects these two buses. Both AHB and APB Buses operate at the same or half the clock rate of processor core.

The APB Bridge is the only bus master residing on the APB bus. All APB slaves are mapped onto memory block MB8 in the MCU 32-bit addressing space. A central address decoder is implemented inside the bridge to generate select signals for individual peripherals. In addition, since the base address of each APB slave is associated with select signals, the address bus on APB contains only the value of offset address.

The maximum address space that can be allocated to a single APB slave is 64 KB, i.e. 16-bit address lines. The width of the data bus is mainly constrained to 32 bits. In the case where an APB slave needs large amount of transfers, the device driver can also request DMA channels to conduct a burst of data transfer. The base address and data width of each peripheral are listed in **Table 54**.

Table 54 ***Register Base Addresses for MCU Peripherals***

Address	Description	Data Width	Software Base ID
8001_0000h	Configuration Registers (Clock, Power Down and Reset)	32	CONFGSYS_CONFG_BASE
8002_0000h	General Purpose Inputs/Outputs	16	MDGPIO_BASE
8003_0000h	Reset Generation Unit	16	MDRGU_BASE
8100_0000h	External Memory Interface	32	EMI_BASE
8101_0000h	Interrupt Controller	32	MDCIRQ_BASE
8102_0000h	DMA Controller	32	MDDMA_BASE
8103_0000h	UART 1	16	UART1_BASE
8104_0000h	UART 2	16	UART2_BASE
8105_0000h	UART 3	16	UART3_BASE
8106_0000h	General Purpose Timer	16	MDGPT_BASE
810A_0000h	SIM	16	SIM_BASE
8112_0000h	NAND Flash Interface	32	NFI_BASE
8116_0000h	CCIF	32	CCIF_BASE
8119_0000h	MCU-DSP Shared Register 1	16	SHARE1_BASE
811B_0000h	Log Accelerator	32	LA_BASE

 © 2009 MediaTek Inc.

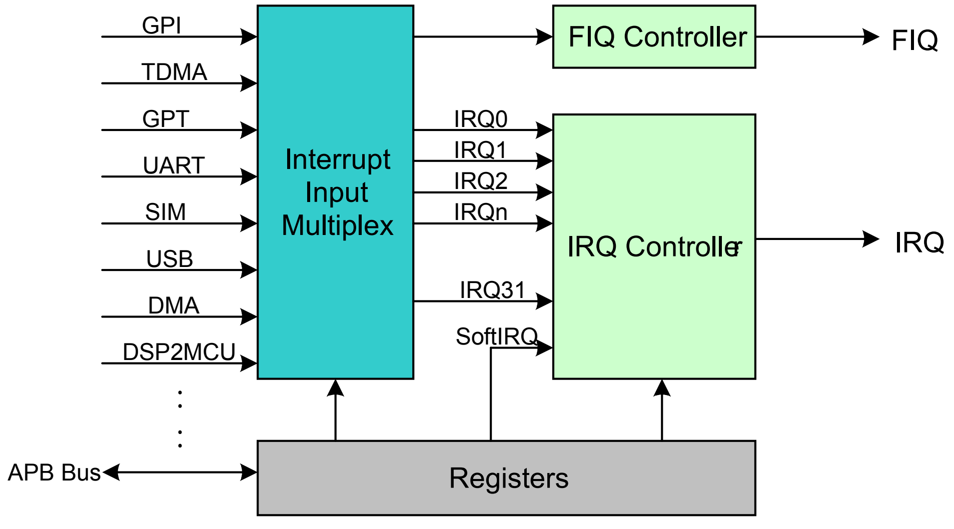
811C_0000h	MDMCUSYS configuration register	32	MDMCUSYS_CONFG_BASE
811D_0000h	UART 4	16	UART4_BASE
811E_0000h	MCU-DSP Shared Register 2	16	SHARE2_BASE
811F_0000h	TDMA Timer	32	TDMA_BASE
8120_0000h	Automatic Frequency Control Unit	13	AFC_BASE
8121_0000h	SIM2	16	SIM2_BASE
8122_0000h	NFIECC	32	NFIECC_BASE
8200_0000h	MD2GSYS configuration register	32	MD2GSYS_CONFG_BASE
8201_0000h	Baseband Serial Interface	32	BSI_BASE
8202_0000h	Baseband Parallel Interface	16	BPI_BASE
8204_0000h	Automatic Power Control Unit	32	APC_BASE
8206_0000h	Divider/Modulus Coprocessor	32	DIVIDER_BASE
8207_0000h	Frame Check Sequence	16	FCS_BASE
8208_0000h	GPRS Cipher Unit	32	GCU_BASE
8209_0000h	CSD Format Conversion Coprocessor	32	CSD_ACC_BASE
820A_0000h	MCU-DSP Shared Register 1	16	SHARED1_BASE
820B_0000h	IRDBG1	16	IRDBG1_BASE
820C_0000h	MCU-DSP Shared Register 2	16	SHARED2_BASE
820D_0000h	IRDBG2	16	IRDBG2_BASE
820E_0000h	DSP Patch Unit	16	PATCH_BASE
820F_0000h	Video Front End	32	VFE_BASE
8210_0000h	Baseband Front End	16	BFE_BASE
8301_0000h	Analog Interface Controller	16	MIXED_BASE

3.4 Interrupt Controller

3.4.1 General Description

Figure 47 outlines the major functionality of the MCU Interrupt Controller. The interrupt controller processes all interrupt sources coming from external lines and internal MCU peripherals. Since ARM7EJ-S core supports two levels of interrupt latency, this controller generates two request signals: FIQ for fast, low latency interrupt request and IRQ for more general interrupts with lower priority.

© 2009 MediaTek Inc.



● *Figure 48* ***Block Diagram of the Interrupt Controller***

One and only one of the interrupt sources can be assigned to FIQ Controller and have the highest priority in requesting timing critical service. All the others share the same IRQ signal by connecting them to IRQ Controller. The IRQ Controller manages up 32 interrupt lines of IRQ0 to IRQ31 with fixed priority in descending order.

The Interrupt Controller provides a simple software interface by mean of registers to manipulate the interrupt request shared system. IRQ Selection Registers and FIQ Selection Register determine the source priority and connecting relation among sources and interrupt lines. IRQ Source Status Register allows software program to identify the source of interrupt that generates the interrupt request. IRQ Mask Register provides software to mask out undesired sources some time. End of Interrupt Register permits software program to indicate to the controller that a certain interrupt service routine has been finished.

Binary coded version of IRQ Source Status Register is also made available for software program to helpfully identify the interrupt source. Note that while taking advantage of this feature, it should also take the binary coded version of End of Interrupt Register coincidently.

The essential Interrupt Table of ARM7EJ-S core is shown as Table **55**.

Address	Description
00000000h	System Reset
00000018h	IRQ
0000001Ch	FIQ

Table 56 ***Interrupt Table of ARM7EJ-S***

3.4.1.1 Interrupt Source Masking

Interrupt controller provides the function of Interrupt Source Masking by the way of programming MASK register. Any of them can be masked individually.

 © 2009 MediaTek Inc.

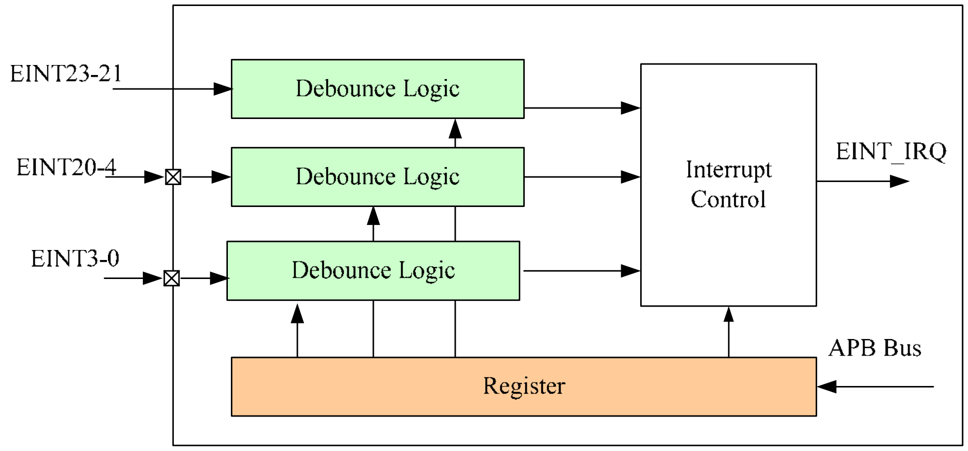
However, because of the bus latency, the masking takes effect no earlier than 3 clock cycles later. In this time, the to-be-masked interrupts could come in and generate an IRQ pulse to MCU, and then disappear immediately. This IRQ forces MCU going to Interrupt Service Routine and polling Status Register (IRQ_STA(IRQ_STAH+IRQ_STAL) or IRQ_STA2), but the register shows there is no interrupt. This might cause MCU malfunction.
There are two ways for programmer to protect their software.

1. Return from ISR (Interrupt Service Routine) immediately while the Status register shows no interrupt.
2. Set I bit of MCU before doing Interrupt Masking, and then clear it after Interrupt Masking done.

Both avoid the problem, but the first item recommended to have in the ISR.

3.4.1.2 External Interrupt

This interrupt controller also integrates an External Interrupt Controller that can support up to 21 interrupt requests coming from external sources, the EINT0~20, and 3 WakeUp interrupt requests, i.e. EINT21~23, coming from peripherals used to inform system to resume the system clock. EINT0~EINT4 interrupt source can be configured as from external pin or internal peripherals.
The external interrupts can be used for different kind of applications, mainly for event detections: detection of hand free connection, detection of hood opening, detection of battery charger connection.
Since the external event may be unstable in a certain period, a de-bounce mechanism is introduced to ensure the functionality. The circuitry is mainly used to verify that the input signal remains stable for a programmable number of periods of the clock. When this condition is satisfied, for the appearance or the disappearance of the input, the output of the de-bounce logic changes to the desired state. Note that, because it uses the 32 KHz slow clock for performing the de-bounce process, the parameter of de-bounce period and de-bounce enable takes effect no sooner than one 32 KHz clock cycle (~31.25us) after the software program sets them. When the sources of External Interrupt Controller are used to resume the system clock in sleep mode, the de-bounce mechanism must be enabled. However, the polarities of EINTs are clocked with the system clock. Any changes to them take effect immediately.



- *Figure 49* ***Block Diagram of External Interrupt Controller***

© 2009 MediaTek Inc.

3.4.1.3 External Interrupt Input Pins

EINT	Edge / Level HW Debounce	SOURCE PIN	SUPPLEMENT
EINT0	Edge / Level Yes	If (GPIO59 M==1) then EINT0= USB DP Pin else EINT0= GPIO59	1. GPIOs should be in the input mode and are effected by GPIO data input inversion registers. 2. GPIOxx_M is the GPIO mode control registers, please refer to GPIO segment.
EINT1	Edge / Level Yes	If (GPIO60 M==1) then EINT1= UART1_RXD else EINT1= GPIO60	
EINT2	Edge / Level Yes	If (GPIO61 M==1) then EINT2= URXD2 else EINT2= GPIO61	
EINT3	Edge / Level Yes	If (GPIO62 M==1) then EINT3 = URXD3 else EINT3= GPIO62	
EINT4	Edge / Level Yes	if(GPIO63_M==1) then EINT4=GPIO63 else EINT4=1	
EINT5	Edge / Level Yes	if(GPIO64_M==1) then EINT5=GPIO64 else EINT5=1	
EINT6	Edge / Level Yes	if(GPIO65_M==1) then EINT6=GPIO65 else EINT6=1	
EINT7	Edge / Level Yes	if(GPIO66_M==1) then EINT7=GPIO66 else EINT7=1	
EINT8	Edge / Level Yes	if(GPIO21_M==1) then EINT8=GPIO21 else EINT8=1	
EINT9	Edge / Level Yes	if(GPIO22_M==3) then EINT9=GPIO22 else EINT9=1	
EINT10	Edge / Level Yes	if(GPIO87_M==3) then EINT10=GPIO87 else EINT10=1	
EINT11	Edge / Level Yes	if(GPIO88_M==3) then EINT11=GPIO88 else EINT11=1	
EINT12	Edge / Level Yes	if(GPIO89_M==3) then EINT12=GPIO89 else EINT12=1	
EINT1	Edge / Level Yes	if(GPIO90_M==3) then EINT13=GPIO90 else EINT13=1	

3			
EINT14	Edge / Level Yes	if(GPIO91_M==3) then EINT14=GPIO91 else EINT14=1	
EINT15	Edge / Level Yes	if(GPIO1_M==3) then EINT15=GPIO1 else EINT15=1	
EINT16	Edge / Level Yes	if(GPIO54_M==2) then EINT16=GPIO54 else EINT16=1	
EINT17	Edge / Level Yes	if(GPIO55_M==3) then EINT17=GPIO55 else EINT17=1	
EINT18	Edge / Level Yes	if(GPIO56_M==2) then EINT18=GPIO56 else EINT18=1	
EINT19	Edge / Level Yes	if(GPIO57_M==2) then EINT19=GPIO57 else EINT19=1	
EINT20	Edge / Level Yes	if(GPIO58_M==2) then EINT20=GPIO58 else EINT20=1	
EINT21	Edge / Level Yes	USB20 IDDIG	
EINT22	Edge / Level Yes	USB20 VBUSVALID	
EINT23	Edge / Level Yes	CPU Interface IRQ_B	

REGISTER ADDRESS	REGISTER NAME	SYNONYM
MDCIRQ + 0000h	IRQ Selection 0 Register	MDIRQ_SEL0
MDCIRQ + 0004h	IRQ Selection 1 Register	MDIRQ_SEL1
MDCIRQ + 0008h	IRQ Selection 2 Register	MDIRQ_SEL2
MDCIRQ + 000Ch	IRQ Selection 3 Register	MDIRQ_SEL3
MDCIRQ + 0010h	IRQ Selection 4 Register	MDIRQ_SEL4
MDCIRQ + 0014h	IRQ Selection 5 Register	MDIRQ_SEL5

MDCIRQ + 0018h	FIQ Selection Register	MDFIQ_SEL
MDCIRQ + 001Ch	IRQ Mask Register	MDIRQ_MASK
MDCIRQ + 0020h	IRQ Mask Disable Register	MDIRQ_MASK_DIS
MDCIRQ + 0024h	IRQ Mask Enable Register	MDIRQ_MASK_EN
MDCIRQ + 0028h	IRQ Status Register	MDIRQ_STA
MDCIRQ + 002Ch	IRQ End of Interrupt Register	MDIRQ_EOI
MDCIRQ + 0030h	IRQ Sensitive Register	MDIRQ_SENS
MDCIRQ + 0034h	IRQ Software Interrupt Register	MDIRQ_SOFT
MDCIRQ + 0038h	FIQ Control Register	MDFIQ_CON
MDCIRQ + 003Ch	FIQ End of Interrupt Register	MDFIQ_EOI
MDCIRQ + 0040h	Binary Coded Value of IRQ_STATUS	MDIRQ_STA2
MDCIRQ + 0044h	Binary Coded Value of IRQ_EOI	MDIRQ_EOI2
MDCIRQ + 0100h	EINT Status Register	MDEINT_STA
MDCIRQ + 0104h	EINT Mask Register	MDEINT_MASK
MDCIRQ + 0108h	EINT Mask Disable Register	MDEINT_MASK_DIS
MDCIRQ + 010Ch	EINT Mask Enable Register	MDEINT_MASK_EN
MDCIRQ + 0110h	EINT Interrupt Acknowledge Register	MDEINT_INTACK
MDCIRQ + 0114h	EINT Sensitive Register	MDEINT_SENS
MDCIRQ + 0120h	EINT0 De-bounce Control Register	MDEINT0_CON
MDCIRQ + 0130h	EINT1 De-bounce Control Register	MDEINT1_CON
MDCIRQ + 0140h	EINT2 De-bounce Control Register	MDEINT2_CON
MDCIRQ + 0150h	EINT3 De-bounce Control Register	MDEINT3_CON
MDCIRQ + 0160h	EINT4 De-bounce Control Register	MDEINT4_CON
MDCIRQ + 0170h	EINT5 De-bounce Control Register	MDEINT5_CON
MDCIRQ + 0180h	EINT6 De-bounce Control Register	MDEINT6_CON
MDCIRQ + 0190h	EINT7 De-bounce Control Register	MDEINT7_CON
MDCIRQ + 01a0h	EINT8 De-bounce Control Register	MDEINT8_CON
MDCIRQ + 01b0h	EINT9 De-bounce Control Register	MDEINT9_CON
MDCIRQ + 01c0h	EINT10 De-bounce Control Register	MDEINT10_CON
MDCIRQ + 01d0h	EINT11 De-bounce Control Register	MDEINT11_CON
MDCIRQ + 01e0h	EINT12 De-bounce Control Register	MDEINT12_CON
MDCIRQ + 01f0h	EINT13 De-bounce Control Register	MDEINT13_CON
MDCIRQ + 0200h	EINT14 De-bounce Control Register	MDEINT14_CON
MDCIRQ + 0210h	EINT15 De-bounce Control Register	MDEINT15_CON
MDCIRQ + 0220h	EINT16 De-bounce Control Register	MDEINT16_CON
MDCIRQ + 0230h	EINT17 De-bounce Control Register	MDEINT17_CON
MDCIRQ + 0240h	EINT18 De-bounce Control Register	MDEINT18_CON

MDCIRQ + 0250h	EINT19 De-bounce Control Register	MDEINT19_CON
MDCIRQ + 0260h	EINT20 De-bounce Control Register	MDEINT20_CON
MDCIRQ + 0270h	EINT21 De-bounce Control Register	MDEINT21_CON
MDCIRQ + 0280h	EINT22 De-bounce Control Register	MDEINT22_CON
MDCIRQ + 0290h	EINT23 De-bounce Control Register	MDEINT23_CON

Table 57 ***Interrupt Controller Register Map***

3.4.2 Register Definitions

MDCIRQ+0000h IRQ Selection 0 Register MDIRQ_SEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ5**					**IRQ4**					**IRQ3**			
Type			R/W					R/W					R/W			
Reset			5					4					3			
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		**IRQ2**					**IRQ1**					**IRQ0**				
Type		R/W					R/W					R/W				
Reset		2					1					0				

MDCIRQ+0004h IRQ Selection 1 Register MDIRQ_SEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQB**					**IRQA**					**IRQ9**			
Type			R/W					R/W					R/W			
Reset			B					A					9			
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		**IRQ8**					**IRQ7**					**IRQ6**				
Type		R/W					R/W					R/W				
Reset		8					7					6				

MDCIRQ+0008h IRQ Selection 2 Register MDIRQ_SEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**IRQ11**					**IRQ10**					**IRQF**			
Type			R/W					R/W					R/W			
Reset			11					10					F			
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		**IRQE**					**IRQD**					**IRQC**				
Type		R/W					R/W					R/W				
Reset		E					D					C				

MDCIRQ+000Ch IRQ Selection 3 Register MDIRQ_SEL3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name			IRQ17					IRQ16					IRQ15			
Type			R/W					R/W					R/W			
Reset			17					16					15			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		IRQ14					IRQ13					IRQ12				
Type		R/W					R/W					R/W				
Reset		14					13					12				

MDCIRQ+0010h IRQ Selection 4 Register MDIRQ_SEL4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			IRQ1D					IRQ1C					IRQ1B			
Type			R/W					R/W					R/W			
Reset			1D					1C					1B			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		IRQ1A					IRQ19					IRQ18				
Type		R/W					R/W					R/W				
Reset		1A					19					18				

MDCIRQ+0014h IRQ Selection 5 Register MDIRQ_SEL5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							IRQ1F					IRQ1E				
Type							R/W					R/W				
Reset							1F					1E				

MDCIRQ+0018h FIQ Selection Register MDFIQ_SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIQ				
Type												R/W				
Reset												0				

The IRQ/FIQ Selection Registers provide system designers with a flexible routing scheme to make various mappings of priority among interrupt sources possible. The registers allow the interrupt sources to be mapped onto interrupt requests of either FIQ or IRQ. While only one interrupt source can be assigned to FIQ, the other ones share IRQs by mapping them onto IRQ0 to IRQ1F connected to IRQ controller. The priority sequence of IRQ0~IRQ1F is fixed, i.e. IRQ0 > IRQ1 > IRQ2 > … > IRQ1E > IRQ1F. During the software configuration process, the Interrupt Source Code of desired interrupt source should be written into source

field of the corresponding IRQ_SEL0-IRQ_SEL4/FIQ_SEL. 5-bit Interrupt Source Codes for all interrupt sources are fixed and defined.

Interrupt Source	STA2 (Hex)	STA
GPI_FIQ	0	00000001
TDMA_CTIRQ1	1	00000002
TDMA_CTIRQ2	2	00000004
DSP12CPU	3	00000008
DSP22CPU	4	00000010
SIM1	5	00000020
TDMA	6	00000040
UART1	7	00000080
UART2	8	00000100
UART3	9	00000200
GPTimer	A	00000400
EINT	B	00000800
USB	C	00001000
GPI_MIRQ	D	00002000
WDT	E	00004000
USB1	F	00008000
IRDBG1	10	00010000
IRDBG2	11	00020000
NFI	12	00040000
CCIF	13	00080000
DMA	14	00100000
EMI	15	00200000
UART4	16	00400000
LA	17	00800000
SIM2	18	01000000
NFI_ECC	19	02000000

Table 58 ***Interrupt Source Code for Interrupt Sources***

- **FIQ, IRQ0-1F** The 5-bit content of this field corresponds to an Interrupt Source Code shown above.

MDCIRQ+001Ch IRQ Mask Register MDIRQ_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ1F**	**IRQ1E**	**IRQ1D**	**IRQ1C**	**IRQ1B**	**IRQ1A**	**IRQ19**	**IRQ18**	**IRQ17**	**IRQ16**	**IRQ15**	**IRQ14**	**IRQ13**	**IRQ12**	**IRQ11**	**IRQ10**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQF**	**IRQE**	**IRQD**	**IRQC**	**IRQB**	**IRQA**	**IRQ9**	**IRQ8**	**IRQ7**	**IRQ6**	**IRQ5**	**IRQ4**	**IRQ3**	**IRQ2**	**IRQ1**	**IRQ0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register contains a mask bit for each interrupt line in IRQ Controller. The register allows each interrupt source IRQ0 to IRQ1F to be disabled or masked separately under software control. After a system reset, all bit values are set to 1 to indicate that interrupt requests are prohibited.

- **IRQ0-1F** Mask control for the associated interrupt source in the IRQ controller
  - **0** Interrupt is enabled
  - **1** Interrupt is disabled

MDCIRQ+0020h IRQ Mask Clear Register MDIRQ_MASK_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ1F**	**IRQ1E**	**IRQ1D**	**IRQ1C**	**IRQ1B**	**IRQ1A**	**IRQ19**	**IRQ18**	**IRQ17**	**IRQ16**	**IRQ15**	**IRQ14**	**IRQ13**	**IRQ12**	**IRQ11**	**IRQ10**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQF**	**IRQE**	**IRQD**	**IRQC**	**IRQB**	**IRQA**	**IRQ9**	**IRQ8**	**IRQ7**	**IRQ6**	**IRQ5**	**IRQ4**	**IRQ3**	**IRQ2**	**IRQ1**	**IRQ0**
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

This register is used to clear bits in IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be cleared. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

- **IRQ0-1F** Clear corresponding bits in IRQ Mask Register.
  - **0** No effect
  - **1** Disable the corresponding MASK bit

MDCIRQ+0024h IRQ Mask SET Register MDIRQ_MASK_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ1F**	**IRQ1E**	**IRQ1D**	**IRQ1C**	**IRQ1B**	**IRQ1A**	**IRQ19**	**IRQ18**	**IRQ17**	**IRQ16**	**IRQ15**	**IRQ14**	**IRQ13**	**IRQ12**	**IRQ11**	**IRQ10**
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRQF**	**IRQE**	**IRQD**	**IRQC**	**IRQB**	**IRQA**	**IRQ9**	**IRQ8**	**IRQ7**	**IRQ6**	**IRQ5**	**IRQ4**	**IRQ3**	**IRQ2**	**IRQ1**	**IRQ0**
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S

This register is used to set bits in the IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be set. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

- **IRQ0-1F** Set corresponding bits in IRQ Mask Register.
  - **0** No effect
  - **1** Enable corresponding MASK bit

 © 2009 MediaTek Inc.

MDCIRQ+0028h IRQ Source Status Register MDIRQ_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This Register allows software to poll which interrupt line has generated an IRQ interrupt request. A bit set to 1 indicates a corresponding active interrupt line. Only one flag is active at a time. The IRQ_STA is type of read-clear; write access has no effect on the content.

- **IRQ0-1F** Interrupt indicator for the associated interrupt source.
  - **0** The associated interrupt source is non-active.
  - **1** The associated interrupt source is asserted.

MDCIRQ+002Ch IRQ End of Interrupt Register MDIRQ_EOI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register provides a mean for software to relinquish and to refresh the interrupt controller. Writing a 1 to a specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

- **IRQ0-1F** End of Interrupt command for the associated interrupt line.
  - **0** No service is currently in progress or pending
  - **1** Interrupt request is in-service

MDCIRQ+0030h IRQ Sensitive Register MDIRQ_SENS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All interrupt lines of IRQ Controller, IRQ0~IRQ1F can be programmed as either edge or level sensitive. By default, all the interrupt lines are edge sensitive and should be active LOW. Once a interrupt line is programmed as edge sensitive, an interrupt request is triggered only at the falling edge of interrupt line, and the next interrupt is not accepted until the EOI command is given. However, level sensitive interrupts trigger is according to the signal level of the interrupt line. Once the interrupt line become from HIGH to LOW, an interrupt request is triggered, and another interrupt request is triggered if the signal level remain LOW after an EOI command. Note that in edge sensitive mode, even if the signal level remains LOW after EOI command, another interrupt request is not triggered. That is because edge sensitive interrupt is only triggered at the falling edge.

- **IRQ0-1F** Sensitivity type of the associated Interrupt Source
  - **0** Edge sensitivity with active LOW
  - **1** Level sensitivity with active LOW

MDCIRQ+0034h IRQ Software Interrupt Register MDIRQ_SOFT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ1F	IRQ1E	IRQ1D	IRQ1C	IRQ1B	IRQ1A	IRQ19	IRQ18	IRQ17	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQF	IRQE	IRQD	IRQC	IRQB	IRQA	IRQ9	IRQ8	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting “1” to the specific bit position generates a software interrupt for corresponding interrupt line before mask. This register is used for debug purpose.

- **IRQ0-IRQ1F** Software Interrupt

MDCIRQ+0038h FIQ Control Register MDFIQ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SENS	MASK
Type															R/W	R/W
Reset															0	1

This register provides a means for software program to control the FIQ controller.

- **MASK** Mask control for the FIQ Interrupt Source
  - **0** Interrupt is enabled
  - **1** Interrupt is disabled
- **SENS** Sensitivity type of the FIQ Interrupt Source
  - **0** Edge sensitivity with active LOW

- **1** Level sensitivity with active LOW

MDCIRQ+003Ch FIQ End of Interrupt Register MDFIQ_EOI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**EOI**
Type																WO
Reset																0

This register provides a means for software to relinquish and to refresh the FIQ controller. Writing a '1' to the specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

- **EOI** End of Interrupt command

MDCIRQ+0040h Binary Coded Value of IRQ_STATUS MDIRQ_STA2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**NOIRQ**				**STA**				
Type								RC				RC				
Reset								0				0				

This Register is a binary coded version of IRQ_STA. It is used by the software program to poll which interrupt line has generated the IRQ interrupt request in a much easier way. Any read to it has the same result as reading IRQ_STA. The IRQ_STA2 is also read-only and read-clear; write access has no effect on the content. Note that IRQ_STA2 should be coupled with IRQ_EOI2 while using it.

- **STA** Binary coded value of IRQ_STA
- **NOIRQ** Indicating if there is an IRQ or not. If there is no IRQ, this bit is HIGH, and the value of STA is 0_0000b.

MDCIRQ+0044h Binary Coded Value of IRQ_EOI MDIRQ_EOI2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**EOI**				
Type												WO				
Reset												0				

© 2009 MediaTek Inc.

This register is a binary coded version of IRQ_EOI. It provides an easier way for software program to relinquish and to refresh the interrupt controller. Writing a specific code results in an End of Interrupt command issued internally to the corresponding interrupt line. Note that IRQ_EOI2 should be coupled with IRQ_STA2 while using it.

- **EOI** Binary coded value of IRQ_EOI

MDCIRQ+0100h EINT Interrupt Status Register MDEINT_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									RO	RO	RO	RO	RO	RO	RO	RO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register keeps up with current status of which EINT Source generated the interrupt request. The status will be changed to zero if the corresponding EINT source mask bit is set.

- **EINT0-EINT23** Interrupt Status
  - **0** No interrupt request is generated
  - **1** Interrupt request is pending

MDCIRQ+0104h EINT Interrupt Mask Register MDEINT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register controls whether or not EINT Source is allowed to generate an interrupt request. Setting a "1" to the specific bit position prohibits the external interrupt line from becoming active.

- **EINT0-EINT23** Interrupt Mask
  - **0** Interrupt request is enabled.
  - **1** Interrupt request is disabled.

MDCIRQ+0108h EINT Interrupt Mask Clear Register MDEINT_MASK_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

This register is used to clear individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are cleared (set to 0). Otherwise the interrupt mask bit retains its original value.

- **EINT0-EINT23** Disable mask for the associated external interrupt source
  - **0** No effect.
  - **1** Disable the corresponding MASK bit.

MDCIRQ+010Ch EINT Interrupt Mask Set Register MDEINT_MASK_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S	W1S

This register is used to set individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are set to 1. Otherwise the interrupt mask bit retains its original value.

- **EINT0-EINT23** Disable mask for the associated external interrupt source.
  - **0** No effect.
  - **1** Enable corresponding MASK bit.

MDCIRQ+0110h EINT Interrupt Acknowledge Register MDEINT_INTACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									WO	WO	WO	WO	WO	WO	WO	WO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Writing “1” to the specific bit position is to acknowledge the interrupt request that correspondingly to the external interrupt line source. Write this register to clear edge sensitive EINT triggered status. Write this register to clear EINT edge status first, if the EINT source is changed from level sensitive to edge sensitive.

- **EINT0-EINT23** Interrupt acknowledgement
  - **0** No effect.
  - **1** Interrupt Request is acknowledged.

MDCIRQ+0114h EINT Sensitive Register MDEINT_SENS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									EINT23	EINT22	EINT21	EINT20	EINT19	EINT18	EINT17	EINT16
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Sensitivity type of external interrupt source.

- **EINT0-EINT23** Sensitive type of the associated external interrupt source
  - **0** Edge sensitivity.
  - **1** Level sensitivity.

MDCIRQ+0120h+n*10h EINTn De-bounce Control Register MDEINTn_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN				POL	CNT										
Type	R/W				R/W	R/W										
Reset	0				0	0										

These registers control the de-bounce logic for external interrupt sources in order to minimize the possibility of false activations.

When the external interrupt sources is used to resume the system clock from the sleep mode, the De-bounce control circuit must be enabled.

Note that n is from 0 to 23

- **CNT** De-bounce duration in terms of numbers of 32KHz clock cycles
- **POL** Activation type of the EINT source
  - **0** Negative polarity
  - **1** Positive polarity
- **EN** De-bounce control circuit
  - **0** Disable
  - **1** Enable

 © 2009 MediaTek Inc.

3.5 Direct Memory Access

3.5.1 General Description

A generic DMA Controller is placed on Layer 2 AHB Bus to support fast data transfers and to off-load the processor. With this controller, specific devices on AHB or APB buses can benefit greatly from quick completion of data movement from or to memory modules such as External SRAM, excluding TCM. TCM is invisible for DMA engine. Such Generic DMA Controller can also be used to connect any two devices other than memory module as long as they can be addressed in memory space.

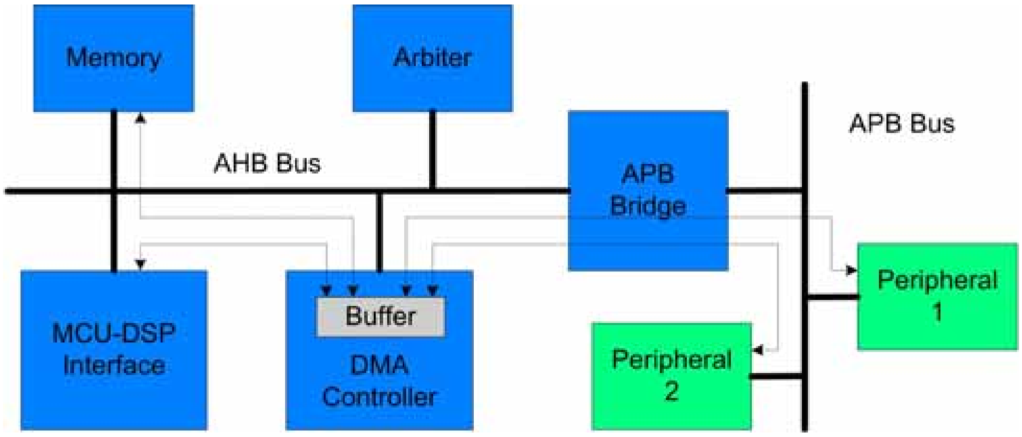


Figure 50 Variety Data Paths of DMA Transfers

Up to six channels of simultaneous data transfers are supported. They are channel 1, 4, 5, and 11 to 14. Each channel has a similar set of registers to be configured to different scheme as desired. If more than six devices are requesting the DMA resources at the same time, software based arbitration should be employed. Once the service candidate is decided, the responsible device driver should configure the Generic DMA Controller properly in order to conduct DMA transfers. Both Interrupt and Polling based schemes in handling the completion event are supported. The block diagram of such generic DMA Controller is illustrated in **Figure 51**.

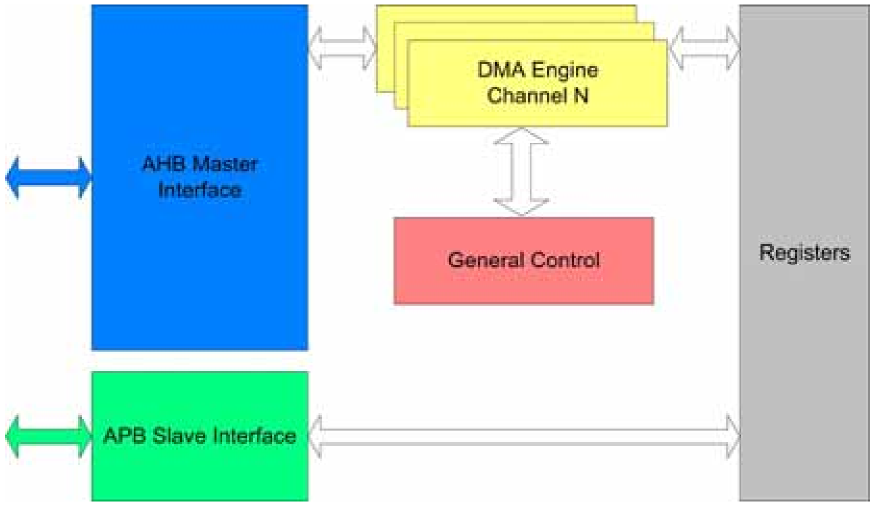


Figure 51 ***Block Diagram of Direct memory Access Module***

3.5.1.1 Full-Size , Half-Size & Half-Size DMA Channels

There are three types of DMA channels in the DMA controller. The first one is called a full-size DMA channel, the second one is called a half-size DMA channel, and the last is Virtual FIFO DMA. Channels 1 is full-size DMA channel; channels 4 and 5 are half-size ones; and channels 11 and 14 are Virtual FIFO DMA channels. The difference between the first two types of DMA channels is that both source and destination address are programmable in full-size DMA channels, but only the address of one side can be programmed in half-size DMA channel. In half-size channels, only either the source or destination address can be programmed, while the addresses of the other side is preset. Which preset address is used depends on the setting of MAS in DMA Channel Control Register. Refer to the Register Definition section for more detail.

3.5.1.2 Ring Buffer & Double Buffer Memory Data Movement

DMA channels 1, 4, 5 support ring-buffer and double-buffer memory data movement. This can be achieved by programming DMA_WPPT and DMA_WPTO, as well as setting WPEN in DMA_CON register to enable. **Figure 52** illustrates how this function works. Once the transfer counter reaches the value of WPPT, the next address will jump to the WPTO address after completing the WPPT data transfer. Note that only one side can be configured as ring-buffer or double-buffer memory, and this is controlled by WPSD in DMA_CON register.

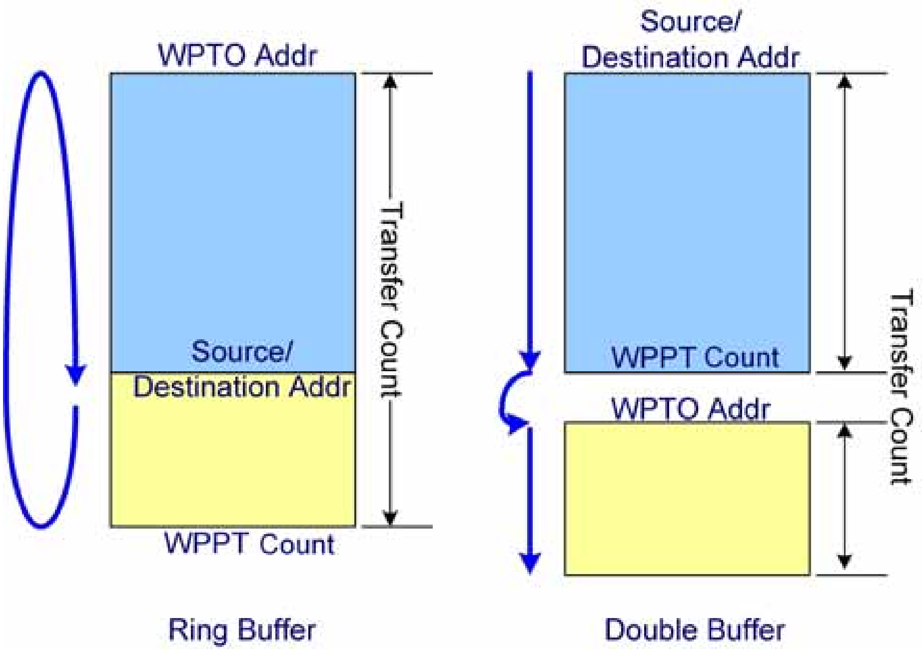


Figure 52 ***Ring Buffer and Double Buffer Memory Data Movement***

3.5.1.3 Unaligned Word Access

The address of word access on AHB bus must be aligned to word boundary, or the 2 LSB is truncated to 00b. If programmers do not notice this, it may cause an incorrect data fetch. In the case where data is to be moved from unaligned addresses to aligned addresses, the word is usually first split into four bytes and then moved byte by byte. This results in four read and four write transfers on the bus.

To improve bus efficiency, unaligned-word access is provided in DMA4, 5. While this function is enabled, DMAs move data from unaligned address to aligned address by executing four continuous byte-read access and one word-write access, reducing the number of transfers on the bus by three.

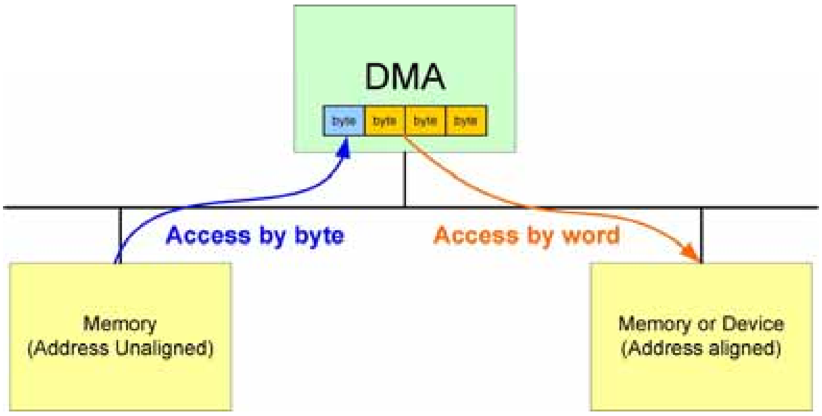


Figure 53 ***Unaligned Word Accesses***

© 2009 MediaTek Inc.

3.5.1.4 Virtual FIFO DMA

Virtual FIFO DMA is used to ease UART control. The difference between the Virtual FIFO DMAs and the ordinary DMAs is that Virtual FIFO DMA contains additional FIFO controller. The read and write pointers are kept in the Virtual FIFO DMA. During a read from the FIFO, the read pointer points to the address of the next data. During a write to the FIFO, the write pointer moves to the next address. If the FIFO is empty, a FIFO read is not allowed. Similarly, data is not written into the FIFO if the FIFO is full. Due to UART flow control requirements, an alert length is programmed. Once the FIFO Space is less than this value, an alert signal is issued to enable UART flow control. The type of flow control performed depends on the setting in UART.

Each Virtual FIFO DMA can be programmed as RX or TX FIFO. This depends on the setting of DIR in DMA_CON register. If DIR is “0”(READ), it means TX FIFO. On the other hand, if DIR is “1”(WRITE), the Virtual FIFO DMA is specified as a RX FIFO.

Virtual FIFO DMA provides an interrupt to MCU. This interrupt informs MCU that there is data in the FIFO, and the amount of data is over or under the value defined in DMA_COUNT register. With this, MCU does not need to poll DMA to know when data must be removed from or put into the FIFO.

Note that Virtual FIFO DMAs cannot be used as generic DMAs, i.e. DMA1, 4, 5.

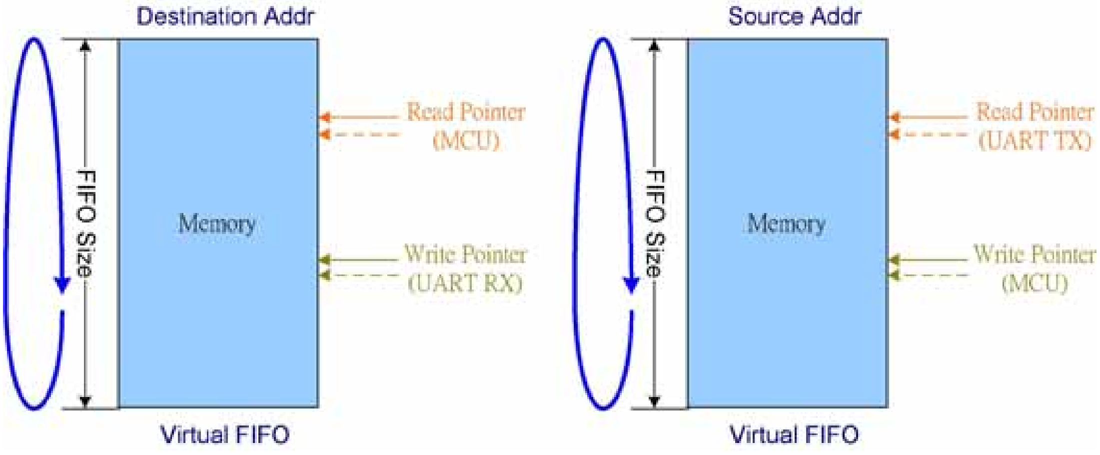


Figure 54 ***Virtual FIFO DMA***

DMA number	Address of Virtual FIFO Access Port	Associated UART
DMA11	6100_0000h	UART1 RX
DMA12	6100_0100h	UART2 RX
DMA13	6200_0200h	UART1 TX
DMA14	6300_0300h	UART2 TX

Table 59 ***Virtual FIFO Access Port***

DMA number	Type	Ring Buffer	Double Buffer	Burst Mode	Unaligned Word Access
DMA1	Full Size	●	●	●	
DMA4	Half Size	●	●	●	●

DMA5	Half Size	●	●	●	●
DMA11	Virtual FIFO	●			
DMA12	Virtual FIFO	●			
DMA13	Virtual FIFO	●			
DMA14	Virtual FIFO	●			

able 60 ***Function List of DMA channels***

REGISTER ADDRESS	REGISTER NAME	SYNONYM
MDDMA + 0000h	DMA Global Status Register	MDDMA_GLBSTA
MDDMA + 0028h	DMA Global Bandwidth Limiter Register	MDDMA_GLBLIMITER
MDDMA + 0100h	DMA Channel 1 Source Address Register	MDDMA1_SRC
MDDMA + 0104h	DMA Channel 1 Destination Address Register	MDDMA1_DST
MDDMA + 0108h	DMA Channel 1 Wrap Point Address Register	MDDMA1_WPPT
MDDMA + 010Ch	DMA Channel 1 Wrap To Address Register	MDDMA1_WPTO
MDDMA + 0110h	DMA Channel 1 Transfer Count Register	MDDMA1_COUNT
MDDMA + 0114h	DMA Channel 1 Control Register	MDDMA1_CON
MDDMA + 0118h	DMA Channel 1 Start Register	MDDMA1_START
MDDMA + 011Ch	DMA Channel 1 Interrupt Status Register	MDDMA1_INTSTA
MDDMA + 0120h	DMA Channel 1 Interrupt Acknowledge Register	MDDMA1_ACKINT
MDDMA + 0124h	DMA Channel 1 Remaining Length of Current Transfer	MDDMA1_RLCT
MDDMA + 0128h	DMA Channel 1 Bandwidth Limiter Register	MDDMA1_LIMITER
MDDMA + 0408h	DMA Channel 4 Wrap Point Address Register	MDDMA4_WPPT
MDDMA + 040Ch	DMA Channel 4 Wrap To Address Register	MDDMA4_WPTO
MDDMA + 0410h	DMA Channel 4 Transfer Count Register	MDDMA4_COUNT
MDDMA + 0414h	DMA Channel 4 Control Register	MDDMA4_CON
MDDMA + 0418h	DMA Channel 4 Start Register	MDDMA4_START
MDDMA + 041Ch	DMA Channel 4 Interrupt Status Register	MDDMA4_INTSTA
MDDMA + 0420h	DMA Channel 4 Interrupt Acknowledge Register	MDDMA4_ACKINT
MDDMA + 0424h	DMA Channel 4 Remaining Length of Current Transfer	MDDMA4_RLCT
MDDMA + 0428h	DMA Channel 4 Bandwidth Limiter Register	MDDMA4_LIMITER
MDDMA + 042Ch	DMA Channel 4 Programmable Address Register	MDDMA4_PGMADDR
MDDMA + 0508h	DMA Channel 5 Wrap Point Address Register	MDDMA5_WPPT
MDDMA + 050Ch	DMA Channel 5 Wrap To Address Register	MDDMA5_WPTO

© 2009 MediaTek Inc.

MDDMA + 0510h	DMA Channel 5 Transfer Count Register	MDDMA5_COUNT
MDDMA + 0514h	DMA Channel 5 Control Register	MDDMA5_CON
MDDMA + 0518h	DMA Channel 5 Start Register	MDDMA5_START
MDDMA + 051Ch	DMA Channel 5 Interrupt Status Register	MDDMA5_INTSTA
MDDMA + 0520h	DMA Channel 5 Interrupt Acknowledge Register	MDDMA5_ACKINT
MDDMA + 0524h	DMA Channel 5 Remaining Length of Current Transfer	MDDMA5_RLCT
MDDMA + 0528h	DMA Channel 5 Bandwidth Limiter Register	MDDMA5_LIMITER
MDDMA + 052Ch	DMA Channel 5 Programmable Address Register	MDDMA5_PGMADDR
MDDMA + 0B10h	DMA Channel 11 Transfer Count Register	MDDMA11_COUNT
MDDMA + 0B14h	DMA Channel 11 Control Register	MDDMA11_CON
MDDMA + 0B18h	DMA Channel 11 Start Register	MDDMA11_START
MDDMA + 0B1Ch	DMA Channel 11 Interrupt Status Register	MDDMA11_INTSTA
MDDMA + 0B20h	DMA Channel 11 Interrupt Acknowledge Register	MDDMA11_ACKINT
MDDMA + 0B28h	DMA Channel 11 Bandwidth Limiter Register	MDDMA11_LIMITER
MDDMA + 0B2Ch	DMA Channel 11 Programmable Address Register	MDDMA11_PGMADDR
MDDMA + 0B30h	DMA Channel 11 Write Pointer	MDDMA11_WRPTR
MDDMA + 0B34h	DMA Channel 11 Read Pointer	MDDMA11_RDPTR
MDDMA + 0B38h	DMA Channel 11 FIFO Count	MDDMA11_FFCNT
MDDMA + 0B3Ch	DMA Channel 11 FIFO Status	MDDMA11_FFSTA
MDDMA + 0B40h	DMA Channel 11 Alert Length	MDDMA11_ALTLEN
MDDMA + 0B44h	DMA Channel 11 FIFO Size	MDDMA11_FFSIZE
MDDMA + 0C10h	DMA Channel 12 Transfer Count Register	MDDMA12_COUNT
MDDMA + 0C14h	DMA Channel 12 Control Register	MDDMA12_CON
MDDMA + 0C18h	DMA Channel 12 Start Register	MDDMA12_START
MDDMA + 0C1Ch	DMA Channel 12 Interrupt Status Register	MDDMA12_INTSTA
MDDMA + 0C20h	DMA Channel 12 Interrupt Acknowledge Register	MDDMA12_ACKINT
MDDMA + 0C28h	DMA Channel 12 Bandwidth Limiter Register	MDDMA12_LIMITER
MDDMA + 0C2Ch	DMA Channel 12 Programmable Address Register	MDDMA12_PGMADDR
MDDMA + 0C30h	DMA Channel 12 Write Pointer	MDDMA12_WRPTR
MDDMA + 0C34h	DMA Channel 12 Read Pointer	MDDMA12_RDPTR
MDDMA + 0C38h	DMA Channel 12 FIFO Count	MDDMA12_FFCNT
MDDMA + 0C3Ch	DMA Channel 12 FIFO Status	MDDMA12_FFSTA

© 2009 MediaTek Inc.

MDDMA + 0C40h	DMA Channel 12 Alert Length	MDDMA12_ALTLEN
MDDMA + 0C44h	DMA Channel 12 FIFO Size	MDDMA12_FFSIZE
MDDMA + 0D14h	DMA Channel 13 Control Register	MDDMA13_CON
MDDMA + 0D18h	DMA Channel 13 Start Register	MDDMA13_START
MDDMA + 0D1Ch	DMA Channel 13 Interrupt Status Register	MDDMA13_INTSTA
MDDMA + 0D20h	DMA Channel 13 Interrupt Acknowledge Register	MDDMA13_ACKINT
MDDMA + 0D28h	DMA Channel 13 Bandwidth Limiter Register	MDDMA13_LIMITER
MDDMA + 0D2Ch	DMA Channel 13 Programmable Address Register	MDDMA13_PGMADDR
MDDMA + 0D30h	DMA Channel 13 Write Pointer	MDDMA13_WRPTR
MDDMA + 0D34h	DMA Channel 13 Read Pointer	MDDMA13_RDPTR
MDDMA + 0D38h	DMA Channel 13 FIFO Count	MDDMA13_FFCNT
MDDMA + 0D3Ch	DMA Channel 13 FIFO Status	MDDMA13_FFSTA
MDDMA + 0D40h	DMA Channel 13 Alert Length	MDDMA13_ALTLEN
MDDMA + 0D44h	DMA Channel 13 FIFO Size	MDDMA13_FFSIZE
MDDMA + 0E14h	DMA Channel 14 Control Register	MDDMA14_CON
MDDMA + 0E18h	DMA Channel 14 Start Register	MDDMA14_START
MDDMA + 0E1Ch	DMA Channel 14 Interrupt Status Register	MDDMA14_INTSTA
MDDMA + 0E20h	DMA Channel 14 Interrupt Acknowledge Register	MDDMA14_ACKINT
MDDMA + 0E28h	DMA Channel 14 Bandwidth Limiter Register	MDDMA14_LIMITER
MDDMA + 0E2Ch	DMA Channel 14 Programmable Address Register	MDDMA14_PGMADDR
MDDMA + 0E30h	DMA Channel 14 Write Pointer	MDDMA14_WRPTR
MDDMA + 0E34h	DMA Channel 14 Read Pointer	MDDMA14_RDPTR
MDDMA + 0E38h	DMA Channel 14 FIFO Count	MDDMA14_FFCNT
MDDMA + 0E3Ch	DMA Channel 14 FIFO Status	MDDMA14_FFSTA
MDDMA + 0E40h	DMA Channel 14 Alert Length	MDDMA14_ALTLEN
MDDMA + 0E44h	DMA Channel 14 FIFO Size	MDDMA14_FFSIZE

Table 61 ***DMA Controller Register Map***

3.5.2 Register Definitions

Register programming tips:

- Start registers shall be cleared, when associated channels are being programmed.
- PGMADDR, i.e. programmable address, only exists in half-size DMA channels. If DIR in Control Register is high, PGMADDR represents Destination Address. Conversely, If DIR in Control Register is low, PGMADDR represents Source Address.

 © 2009 MediaTek Inc.

- Functions of ring-buffer and double-buffer memory data movement can be activated on either source side or destination side by programming DMA_WPPT & and DMA_WPTO, as well as setting WPEN in DMA_CON register high. WPSD in DMA_CON register determines the activated side.

MDDMA+0000h DMA Global Status Register MDDMA_GLBSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**IT14**	**RUN14**	**IT13**	**RUN13**	**IT12**	**RUN12**	**IT11**	**RUN11**				
Type					RO	RO	RO	RO	RO	RO	RO	RO				
Reset					0	0	0	0	0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**IT5**	**RUN5**	**IT4**	**RUN4**					**IT1**	**RUN1**
Type							RO	RO	RO	RO					RO	RO
Reset							0	0	0	0					0	0

This register helps software program keep track of the global status of DMA channels.

RUN_N DMA channel n status

- **0** Channel n is stopped or has completed the transfer already.
- **1** Channel n is currently running.

IT_N Interrupt status for channel n

- **0** No interrupt is generated.

2 An interrupt is pending and waiting for service.

MDDMA+0028h DMA Global Bandwidth limiter Register MDDMA_GLBLIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**GLBLIMITER**							
Type									WO							
Reset									0							

- Please refer to the expression in DMAn_LIMITER for detailed note. The value of DMA_GLBLIMITER is set to all DMA channels, 1, 4, 5, 11, 12.

MDDMA+0n00h DMA Channel n Source Address Register MDDMAn_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SRC[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC[15:0]**															
Type	R/W															
Reset	0															

© 2009 MediaTek Inc.

The above registers contain the base or current source address that the DMA channel is currently operating on. Writing to this register specifies the base address of transfer source for a DMA channel. Before programming these registers, the software program should make sure that STR in DMAn_START is set to 0; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value from which the DMA is reading.

Note that n is 1 and SRC can't be TCM address. TCM is not accessible by DMA..

SRC **SRC**[31:0] specifies the base or current address of transfer source for a DMA channel, i.e. channel 1.

- **WRITE** Base address of transfer source
- **READ** Address from which DMA is reading

MDDMA+0n04h DMA Channel n Destination Address Register MDDMAn_DST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DST[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST[15:0]															
Type	R/W															
Reset	0															

The above registers contain the base or current destination address that the DMA channel is currently operating on.. Writing to this register specifies the base address of the transfer destination for a DMA channel. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0'; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value to which the DMA is writing.

Note that n is 1 and DST can't be TCM address. TCM is not accessible by DMA.

DST **DST**[31:0] specifies the base or current address of transfer destination for a DMA channel, i.e. channel 1.

- **WRITE** Base address of transfer destination.
- **READ** Address to which DMA is writing.

MDDMA+0n08h DMA Channel n Wrap Point Count Register MDDMAn_WPPT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WPPT[15:0]															
Type	R/W															
Reset	0															

The above registers are to specify the transfer count required to perform before the jump point. This can be used to support ring buffer or double buffer style memory accesses. To enable this function, two control bits,

WPEN and WPSD, in DMA control register must be programmed. See the following register description for more details. If the transfercounter in the DMA engine matches this value, an address jump occurs, and the next address is the address specified in DMAn_WPTO. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON is set. Note that the total size of data specify in the wrap point count in a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. WPPT x SIZE.

Note that n is 1, 4, 5.

WPPT **WPPT**[15:0] specifies the amount of the transfer count from start to jumping point for a DMA channel, i.e. channel 1, 4, 5.

- **WRITE** Wrap point transfer count.
- **READ** Value set by the programmer.

MDDMA+0n0Ch DMA Channel n Wrap To Address Register MDDMAn_WPTO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WPTO[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WPTO[15:0]															
Type	R/W															
Reset	0															

The above registers specify the address of the jump destination of a given DMA transfer to support ring buffer or double buffer style memory accesses. To enable this function, set the two control bits, WPEN and WPSD, in the DMA control register. See the following register description for more details. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON should be set.

Note that n is 1, 4, 5

WPTO **WPTO**[31:0] specifies the address of the jump point for a DMA channel, i.e. channel 1, 4, 5.

- **WRITE** Address of the jump destination.
- **READ** Value set by the programmer.

MDDMA+0n10h DMA Channel n Transfer Count Register MDDMAn_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LEN															
Type	R/W															

Reset	0

This register specifies the amount of total transfer count that the DMA channel is required to perform. Upon completion, the DMA channel generates an interrupt request to the processor while ITEN in DMAn_CON is set as '1'. Note that the total size of data being transferred by a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. LEN x SIZE.

For virtual FIFO DMA, this register is used to configure the RX threshold and TX threshold. Interrupt is triggered while FIFO count >= RX threshold in RX path or FIFO count =< TX threshold in TX path. Note that ITEN bit in DMA_CON register shall be set, or no interrupt is issued.

Note that n is 1, 4, 5, 11, 12, 13, 14.

LEN The amount of total transfer count

MDDMA+0n14h DMA Channel n Control Register MDDMAn_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								MAS						DIR	WPEN	WPSD
Type								R/W						R/W	R/W	R/W
Reset								0						0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ITEN					BURST					B2W	DRQ	DINC	SINC	SIZE	
Type	R/W					R/W					R/W	R/W	R/W	R/W	R/W	
Reset	0					0					0	0	0	0	0	

This register contains all the available control schemes for a DMA channel that is ready for software programmer to configure. Note that all these fields cannot be changed while DMA transfer is in progress or an unexpected situation may occur.

Note that n is 1, 4, 5, 11, 12,13,14.

SIZE Data size within the confine of a bus cycle per transfer.
These bits confines the data transfer size between source and destination to the specified value for individual bus cycle. The size is in terms of byte and has maximum value of 4 bytes. It is mainly decided by the data width of a DMA master.

- **00** Byte transfer/1 byte
- **01** Half-word transfer/2 bytes
- **10** Word transfer/4 bytes
- **11** Reserved

SINC Incremental source address. Source addresses increase every transfer. If the setting of SIZE is Byte, Source addresses increase by 1 every single transfer. If Half-Word, increase by 2; and if Word, increase by 4.

- **0** Disable
- **1** Enable

DINC Incremental destination address. Destination addresses increase every transfer. If the setting of SIZE is Byte, Destination addresses increase by 1 every single transfer. If Half-Word, increase by 2; and If Word, increase by 4.

- **0** Disable
- **1** Enable

DREQ Throttle and handshake control for DMA transfer

- **0** No throttle control during DMA transfer or transfers occurred only between memories

2 Hardware handshake management

- The DMA master is able to throttle down the transfer rate by way of request-grant handshake.

B2W Word to Byte or Byte to Word transfer for the applications of transferring non-word-aligned-address data to word-aligned-address data. Note that BURST is set to 4-beat burst while enabling this function, and the SIZE is set to Byte.

NO effect on channel 1, 11, 12,13,14.

- **0** Disable
- **1** Enable

BURST Transfer Type. Burst-type transfers have better bus efficiency. Mass data movement is recommended to use this kind of transfer. However, note that burst-type transfer does not stop until all of the beats in a burst are completed or transfer length is reached. FIFO threshold of peripherals must be configured carefully while being used to move data from/to the peripherals.

What transfer type can be used is restricted by the SIZE. If SIZE is 00b, i.e. byte transfer, all of the four transfer types can be used. If SIZE is 01b, i.e. half-word transfer, 16-beat incrementing burst cannot be used. If SIZE is 10b, i.e. word transfer, only single and 4-beat incrementing burst can be used.

NO effect on channel 17-24

- **000** Single
- **001** Reserved
- **010** 4-beat incrementing burst
- **011** Reserved
- **100** 8-beat incrementing burst
- **101** Reserved
- **110** 16-beat incrementing burst
- **111** Reserved

ITEN DMA transfer completion interrupt enable.

- **0** Disable
- **1** Enable

WPSD The side using address-wrapping function. Only one side of a DMA channel can activate address-wrapping function at a time.

NO effect on channel 17-24

- **0** Address-wrapping on source .
- **1** Address-wrapping on destination.

WPEN Address-wrapping for ring buffer and double buffer. The next address of DMA jumps to WRAP TO address when the current address matches WRAP POINT count.

NO effect on channel 17-24

- **0** Disable
- **1** Enable

 © 2009 MediaTek Inc.

DIR Directions of DMA transfer for half-size and Virtual FIFO DMA channels, i.e. channels 4, 5, 11, 12,13,14. The direction is from the perspective of the DMA masters. WRITE means read from master and then write to the address specified in DMA_PGMADDR, and vice versa.
NO effect on channel 1.

- **0** Read
- **1** Write

MAS Master selection. Specifies which master occupies this DMA channel. Once assigned to certain master, the corresponding DREQ and DACK are connected. For half-size and Virtual FIFO DMA channels, i.e. channels 4, 5, 11, 12,13,14, a predefined address is assigned as well.

- **00000** SIM1
- **00001** UART1 TX
- **00010** UART1 RX
- **00011** UART2 TX
- **00100** UART2 RX
- **00101** NFI TX
- **00110** NFI RX
- **00111** IDMA 1

1000 IDMA 2

- **1001** Reserved
- **1010** Reserved
- **1011** Reserved
- **1100** Reserved
- **1101** SIM2
- **OTHERS** Reserved

MDDMA+0n18h DMA Channel n Start Register MDDMAn_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**STR**															
Type	R/W															
Reset	0															

This register controls the activity of a DMA channel. When STR is changed from 0 to 1, the DMA channel starts to work. Note that prior to setting STR to “1”, all the configurations should be done by giving proper value to the registers. Note also that once the STR is set to “1”, the hardware does not clear it automatically no matter if the DMA channel accomplishes the DMA transfer or not. In other words, the value of STR stays “1” regardless of the completion of DMA transfer. Therefore, the software program should be sure to clear STR to “0” for starting another transfer for the same DMA channel. If this bit is cleared to “0” during DMA transfer is active, software should polling MDDMA_GLBSTA **RUN_N** after this bit is cleared to ensure current DMA transfer is terminated by DMA engine.

Note that n is 1, 4, 5, 11, 12.

MDDMA+0n1Ch DMA Channel n Interrupt Status Register MDDMAn_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INT**															
Type	RO															
Reset	0															

This register shows the interrupt status of a DMA channel. It has the same value as DMA_GLBSTA.

Note that n is 1, 4, 5, 11, 12, 13,14.

INT Interrupt Status for DMA Channel

- **0** No interrupt request is generated.
- **1** One interrupt request is pending and waiting for service.

MDDMA+0n20h DMA Channel n Interrupt Acknowledge Register MDDMAn_ACKINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ACK**															
Type	WO															
Reset	0															

This register is used to acknowledge the current interrupt request associated with the completion event of a DMA channel by software program. Note that this is a write-only register, and any read to it returns a value of "0".

Note that n is 1, 4, 5, 11, 12 ,13,14.

ACK Interrupt acknowledge for the DMA channel

- **0** No effect
- **1** Interrupt request is acknowledged and should be relinquished.

MDDMA+0n24h DMA Channel n Remaining Length of Current Transfer MDDMAn_RLCT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RLCT**															

Type	RO
Reset	0

- This register is to reflect the left transfer count of the transfer. Note that this value is transfer count not the transfer data size.

Note that n is 1, 4, 5.

MDDMA+0n28h DMA Bandwidth limiter Register MDDMAn_LIMITER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**LIMITER**							
Type									R/W							
Reset									0							

- This register is to suppress the Bus utilization of the DMA channel. The value is from 0 to 255. 0 means no limitation, and 255 means totally banned. The value between 0 and 255 means certain DMA can have permission to use AHB every (4 X n) AHB clock cycles.
- Note that it is not recommended to limit the Bus utilization of the DMA channels because this increases the latency of response to the masters, and the transfer rate decreases as well. Before using it, programmer must make sure that the bus masters have some protective mechanism to avoid entering the wrong states.

Note that n is 1, 4, 5, 11, 12.

LIMITER from 0 to 255. 0 means no limitation, 255 means totally banned, and others mean Bus access permission every (4 X n) AHB clock.

MDDMA+0n2Ch DMA Channel n Programmable Address Register MDDMAn_PGMADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PGMADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PGMADDR[15:0]**															
Type	R/W															
Reset	0															

The above registers specify the address for a half-size DMA channel. This address represents a source address if DIR in DMA_CON is set to 0, and represents a destination address if DIR in DMA_CON is set to 1. Before being able to program these register, the software should make sure that STR in DMAn_START is set to ‘0’, that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order.

Note that n is 4, 5, 11, 12, 13,14 and PGMADDR can’t be TCM address. TCM is not accessible by DMA.

PGMADDR **PGMADDR**[31:0] specifies the addresses for a half-size or a Virtual FIFO DMA channel, i.e. channel 4, 5, 11, 12 13 14.

- **WRITE** Base address of transfer source or destination according to DIR bit.
- **READ** Current address of the transfer.

MDDMA+0n30h DMA Channel n Virtual FIFO Write Pointer Register MDDMAn_WRPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WRPTR[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WRPTR[15:0]															
Type	RO															

Note that n is 11, 12.

WRPTR Virtual FIFO Write Pointer.

MDDMA+0n34h DMA Channel n Virtual FIFO Read Pointer Register MDDMAn_RDPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RDPTR[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RDPTR[15:0]															
Type	RO															

Note that n is 11, 12.

RDPTR Virtual FIFO Read Pointer.

MDDMA+0n38h DMA Channel n Virtual FIFO Data Count Register MDDMAn_FFCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FFCNT															
Type	RO															

Note that n is 11, 12.

FFCNT To display the number of data stored in FIFO. 0 means FIFO empty, and FIFO is full if FFCNT is equal to FFSIZE.

MDDMA+0n3Ch DMA Channel n Virtual FIFO Status Register MDDMAn_FFSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														ALT	EMPTY	FULL
Type														RO	RO	RO
Reset														0	1	0

Note that n is 11, 12.

FULL To indicate FIFO is full.

- **0** Not Full

2 Full

EMPTY To indicate FIFO is empty.

- **0** Not Empty

2 Empty

ALT To indicate FIFO Count is larger than ALTLEN. DMA issues an alert signal to UART to enable UART flow control.

- **0** Not reach alert region.

2 Reach alert region.

MDDMA+0n40h DMA Channel n Virtual FIFO Alert Length Register MDDMAn_ALTLEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ALTLEN					
Type											R/W					
Reset											0					

Note that n is 11, 12.

ALTLEN Specifies the Alert Length of Virtual FIFO DMA. Once the remaining FIFO space is less than ALTLEN, an alert signal is issued to UART to enable flow control. Normally, ALTLEN shall be larger than 16 for UART application.

MDDMA+0n44h DMA Channel n Virtual FIFO Size Register MDDMAn_FFSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FFSIZE															
Type	R/W															
Reset	0															

Note that n is 11, 12.13,14

FFSIZE Specifies the FIFO Size of Virtual FIFO DMA.

3.6 General Purpose Inputs/Outputs

MT6516 offers 4 general-purpose I/O pins for modem side usage. By setting the control registers, MCU software can control the direction, the output value, and read the input values on these pins. These GPIOs and are multiplexed with other functionalities to reduce the pin count.

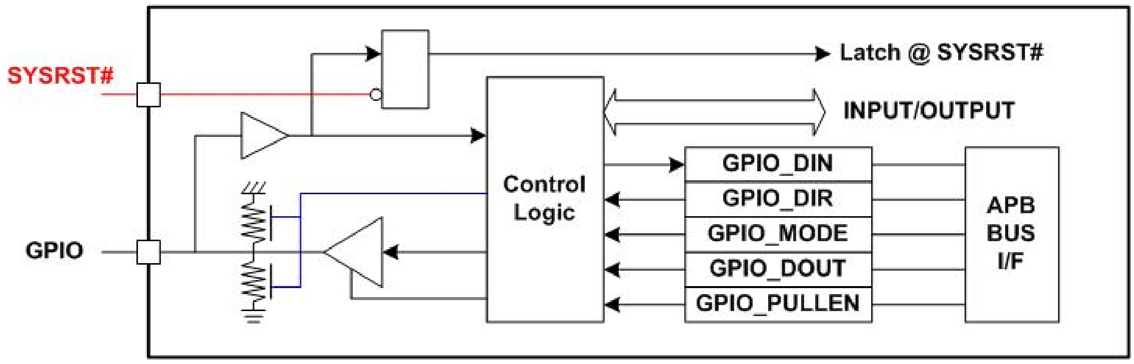


Figure 55 ***GPIO Block Diagram***

GPIOs at RESET

Upon a hardware reset (SYSRST#), GPIOs are all configured as inputs and the following alternative usages of the GPIO pins are enabled.
These GPIOs are used to latch the inputs upon reset to memorize the desired configuration to ensure that the system restarts or boots up in the right mode.

3.6.1 Register Definitions

80020000h GPIO direction control register 1 MD_GPIO_DIR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GPIO3	GPIO2	GPIO1	GPIO0
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

GPIO*n* GPIO direction control

- **0** GPIOs are configured as input
- 2 GPIOs are configured as output

80020100h GPIO pull-up/pull-down enable register 1 MD_GPIO_PULLEN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GPIO3	GPIO2	GPIO1	GPIO0
Type													R/W	R/W	R/W	R/W
Reset													1	1	1	1

GPIO*n* GPIO pull-up/pull-down control

 © 2009 MediaTek Inc.

80020200h GPIO pull-up/pull-down selection register 1 MD_GPIO_PULLSEL1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GPIO3	GPIO2	GPIO1	GPIO0
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0
Note													PD	PD	PD	PD

GPIO*n* GPIO pull-up/pull-down selection control

0 GPIOs pull-down

1 GPIOs pull-up

80020300h GPIO data inversion control register 1 MD_GPIO_DINV1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													INV3	INV2	INV1	INV0
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

INVn GPIO inversion control

0 GPIOs data inversion disable

1 GPIOs data inversion enable

80020400h GPIO data output register 1 MD_GPIO_DOUT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GPIO3	GPIO2	GPIO1	GPIO0
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

GPIOn GPIO data output control

0 GPIOs data output 0

1 GPIOs data output 1

80020500h GPIO data Input register 1 MD_GPIO_DIN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GPIO3	GPIO2	GPIO1	GPIO0
Type													RO	RO	RO	RO
Reset													X	X	X	X

GPIOn GPIOs data input

80020600h GPIO mode control register 1 MD_GPIO_MODE1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									GPIO3_M		GPIO2_M		GPIO1_M		GPIO0_M	

Type					R/W	R/W	R/W	R/W
Reset					00	00	00	00

GPIO0_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: BPI_BUS6
- **10** Reserved
- **11** Reserved

GPIO1_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: BPI_BUS7
- **10** Reserved
- **11** Reserved

GPIO2_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: BPI_BUS8
- **10** Reserved
- **11** Reserved

GPIO3_M GPIO mode selection
- **00** Configured as GPIO function
- **01** O: BPI_BUS9
- **10** Reserved
- **11** Reserved

3.7 General Purpose Timer (MD)

3.7.1 General Description

Three general-purpose timers are provided. The timers are 16 bits long and run independently of each other, although they share the same clock source. Two timers can operate in one of two modes: one-shot mode and auto-repeat mode; the other is a free running timer. In one-shot mode, when the timer counts down and reaches zero, it is halted. In auto-repeat mode, when the timer reaches zero, it simply resets to countdown initial value and repeats the countdown to zero; this loop repeats until the disable signal is set to 1. Regardless of the timer's mode, if the countdown initial value (i.e. MDGPT1_DAT for MDGPT1 or MDGPT_DAT2 for MDGPT2) is written when the timer is running, the new initial value does not take effect until the next time the timer is restarted. In auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the general purpose timer, the desired values for MDGPT_DAT and the MDGPT_PRESCALER registers must first be set.

3.7.2 Register Definitions

MDGPT+0000h **GPT1 Control register** **MDGPT1_CON**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**EN**	**MODE**														
Type	R/W	R/W														
Reset	0	0														

MODE This register controls GPT1 to count repeatedly (in a loop) or just one-shot.

- **0** One-shot mode is selected.
- 2 Auto-repeat mode is selected.

EN This register controls GPT1 to start counting or to stop.

- **0** GPT1 is disabled.
- **1** GPT1 is enabled.

MDGPT+0004h GPT1 Time-Out Interval register MDGPT1_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT [15:0]**															
Type	R/W															
Reset	FFFFh															

CNT [15:0] Initial counting value. GPT1 counts down from GPT1_DAT. When GPT1 counts down to zero, a GPT1 interrupt is generated.

MDGPT+0008h GPT2 Control register MDGPT2_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN**	**MODE**														
Type	R/W	R/W														
Reset	0	0														

MODE This register controls GPT2 to count repeatedly (in a loop) or just one-shot.

- **0** One-shot mode is selected
- **1** Auto-repeat mode is selected

EN This register controls GPT2 to start counting or to stop.

- **0** GPT2 is disabled.
- **1** GPT2 is enabled.

MDGPT+000Ch GPT2 Time-Out Interval register MDGPT2_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT [15:0]**															
Type	R/W															
Reset	FFFFh															

CNT [15:0] Initial counting value. GPT2 counts down from GPT2_DAT. When GPT2 counts down to zero, a GPT2 interrupt is generated.

© 2009 MediaTek Inc.

MDGPT+0010h GPT Status register MDGPT_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**GPT2**	**GPT1**
Type															RC	RC
Reset															0	0

This register illustrates the gptimer timeout status. Each flag is set when the corresponding timer countdown completes, and can be cleared when the CPU reads the status register.

MDGPT+0014h GPT1 Prescaler register MDGPT1_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer1.

000 16384 Hz
001 8192 Hz
010 4096 Hz
011 2048 Hz
102 1024 Hz
103 512 Hz
110 256 Hz
112 128 Hz

MDGPT+0018h GPT2 Prescaler register MDGPT2_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer2.

000 16384 Hz
001 8192 Hz
010 4096 Hz
011 2048 Hz
100 1024 Hz
101 512 Hz
110 256 Hz
112 128 Hz

MDGPT+001Ch GPT3 Control register MDGPT3_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**EN**
Type																R/W
Reset																0

EN This register controls GPT3 to start counting or to stop.

- **0** GPT3 is disabled.
- **1** GPT3 is enabled.

MDGPT+0020h GPT3 Time-Out Interval register MDGPT3_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT[15:0]**															
Type	RO															
Reset	0															

CNT [15:0] If EN=1, GPT3 is a free running timer . Software reads this register for the countdown start value for GPT3.

MDGPT+0024h GPT3 Prescaler register MDGPT3_PRESCALER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PRESCALER [2:0]**		
Type														R/W		
Reset														100b		

PRESCALER This register controls the counting clock for gptimer3.

- **000** 16384 Hz
- **001** 8192 Hz
- **010** 4096 Hz
- **011** 2048 Hz
- **100** 1024 Hz
- **101** 512 Hz
- **110** 256 Hz
- **111** 128 Hz

3.8 L1 Cache controller

3.8.1 General Description

MT6516 core processor has been implemented with a subsystem in which consists of Core Cache and TCM (tightly coupled memory). This subsystem is placed in between MCU core and AHB bus interface, as shown in **Figure 12**.

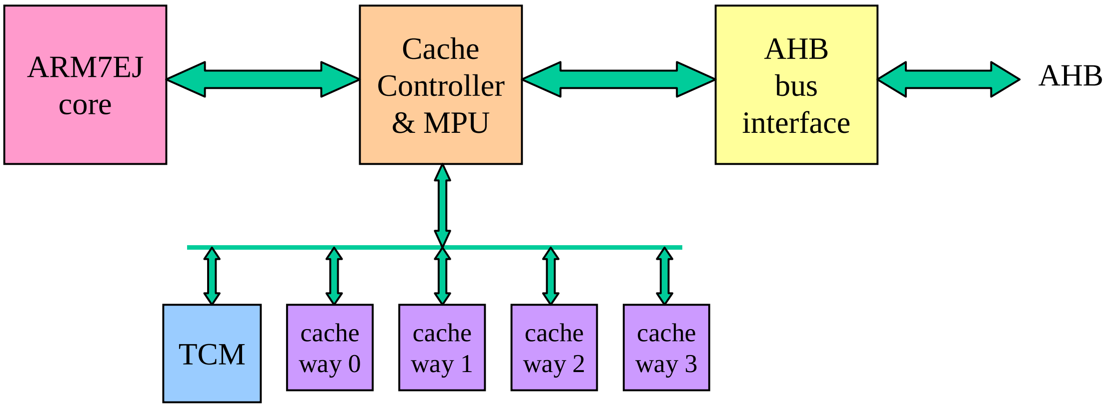


Figure 56 ***[Figure Caption]***

TCM is a high-speed (zero wait state) dedicated memory accessed exclusively only by MCU. Because of the latency penalty when MCU accesses memory or peripherals through on-chip bus, by moving timing critical code and data into TCM, the performance of MCU can be increased and the response to particular events can be guaranteed.

Another method to increase MCU performance is the implementation of cache. In this case, core cache is a small block of memory, in which contents a copy of small portion of cacheable data in the external memory. If MCU reads a cacheable data, the data will be copied into core cache. Once MCU requests the same data again, it can be obtained directly from core cache (called cache hit) instead of fetching it again from external memory. Consider the fact that accessing cache is much faster than accessing external memory through bus system, a faster instruction fetching can be obtained, and that leads to a higher IPC (Instruction Per Cycle) which is a major factor in the evaluation of core performance. Since a large external memory maps to a small cache, cache can hold only a small portion of external memory. If MCU accesses a data not found in the cache (called cache miss), one cache line must be dropped (flushed), the required data and its neighboring data are transferred from external memory to cache (called cache line fill). Before cache line fill, an important step to maintain data consistency between cache and external memory needs to be performed. This important step is so called cache write back, and its introduction will be carried out in the later section. In this design, a cache line consists of eight words (8x32 bits) and it will be introduced later. On the other hand, the best way to utilize TCM is to maintain the critical instruction or data in TCM. This is due to the advantage of TCM that has been described above. After power-on reset, the boot loader copies TCM contents from external storage (such as flash) to internal TCM. If necessary, MCU can replace TCM contents with other data in the external storage during the runtime to implement a mechanism such like “overlay”. TCM is also an ideal place to put stack data.

The sizes of TCM and cache can be set to one of the following 4 configurations:

 © 2009 MediaTek Inc.

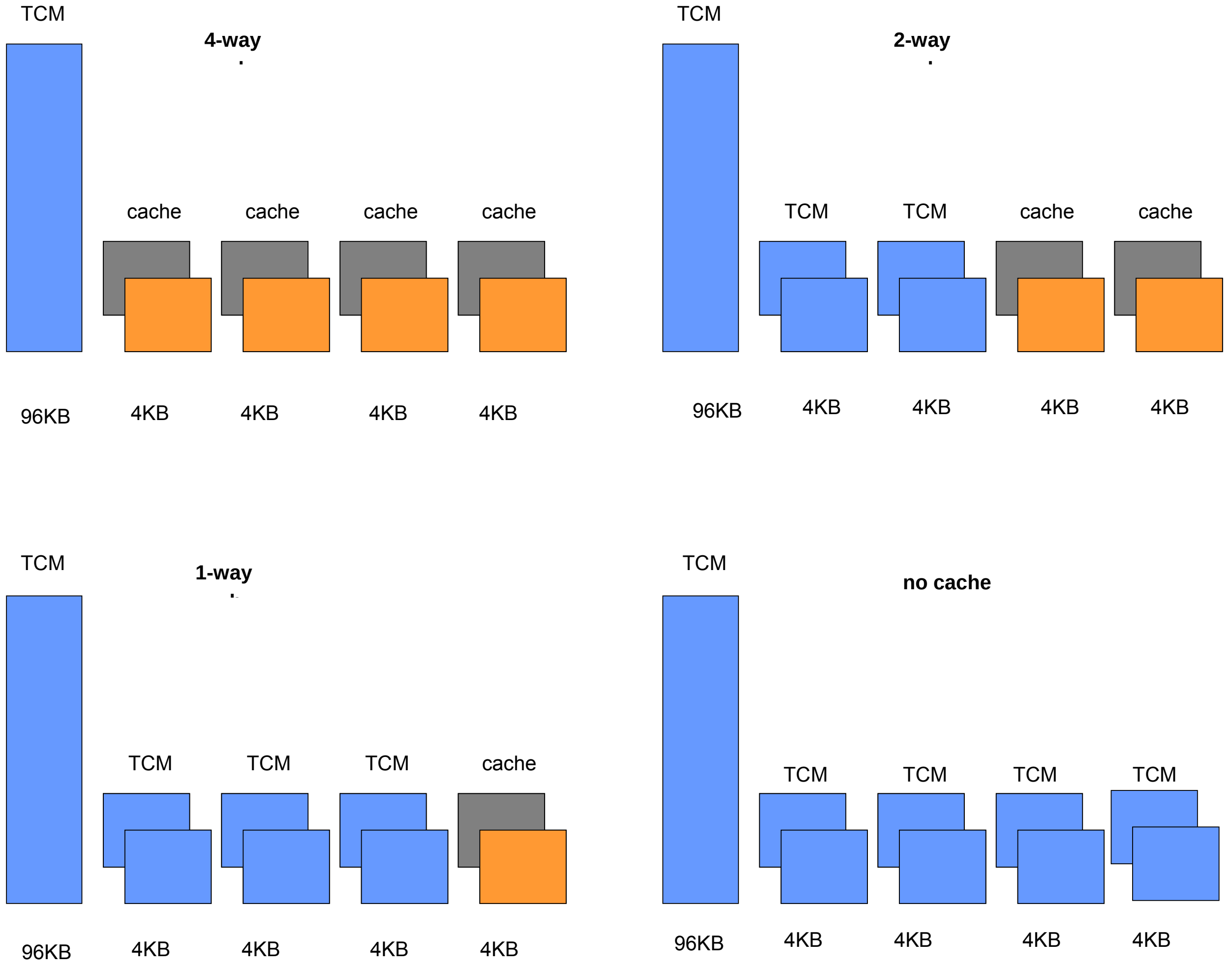


Figure 57 ***[Figure Caption]***

- 96KB TCM, 16KB cache (4 way)
- 104KB TCM, 8KB cache (2 way)
- 108KB TCM, 4KB cache (1 way)
- 112KB TCM, 0KB cache (no cache)

These different configurations provide flexibility for software to adjust and reach optimum system performance. The address mapping of these memories is like the following:

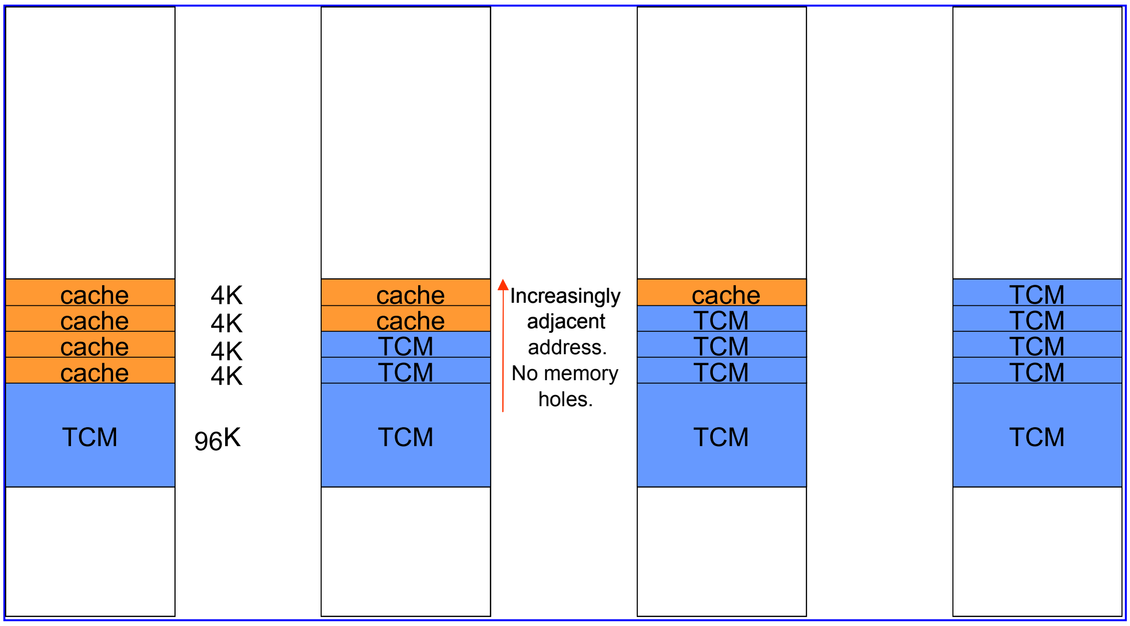


Figure 58 ***[Figure Caption]***

Organization of Cache

The cache system has the following features:

- Write back (in the unit of 4 words)
- Configurable 1/2/4 way set associative (4KB/8KB/16KB)
- Each way has 128 cache lines with 8-word line size (128*8*4=4KB)
- 22-bit tag memory (only 21 of them are used), 1 valid bit for each cache line.
- 2-bit dirty memory (each dirty bit records the dirtiness of half cache line – 4 words)

Each way of cache comprises of two memories: tag memory and data memory. Tag memory stores each line's valid bit and tag (upper 20 bits of address). Data memory stores line data. When MCU accesses memory, the address is compared to the contents of tag memory. First, the line index (address bit [11:5]) is used to locate a line in tag memory. When a particular line is found in the tag memory, the upper 20 bits (address bit [31:12]), called tag, of the desired memory address are compared with the content of found tag line. If an match is found in both line address and tag address plus valid bit is 1, it is said a cache hit, and data from that particular cache way is returned to MCU. This process is illustrated in the following figure:

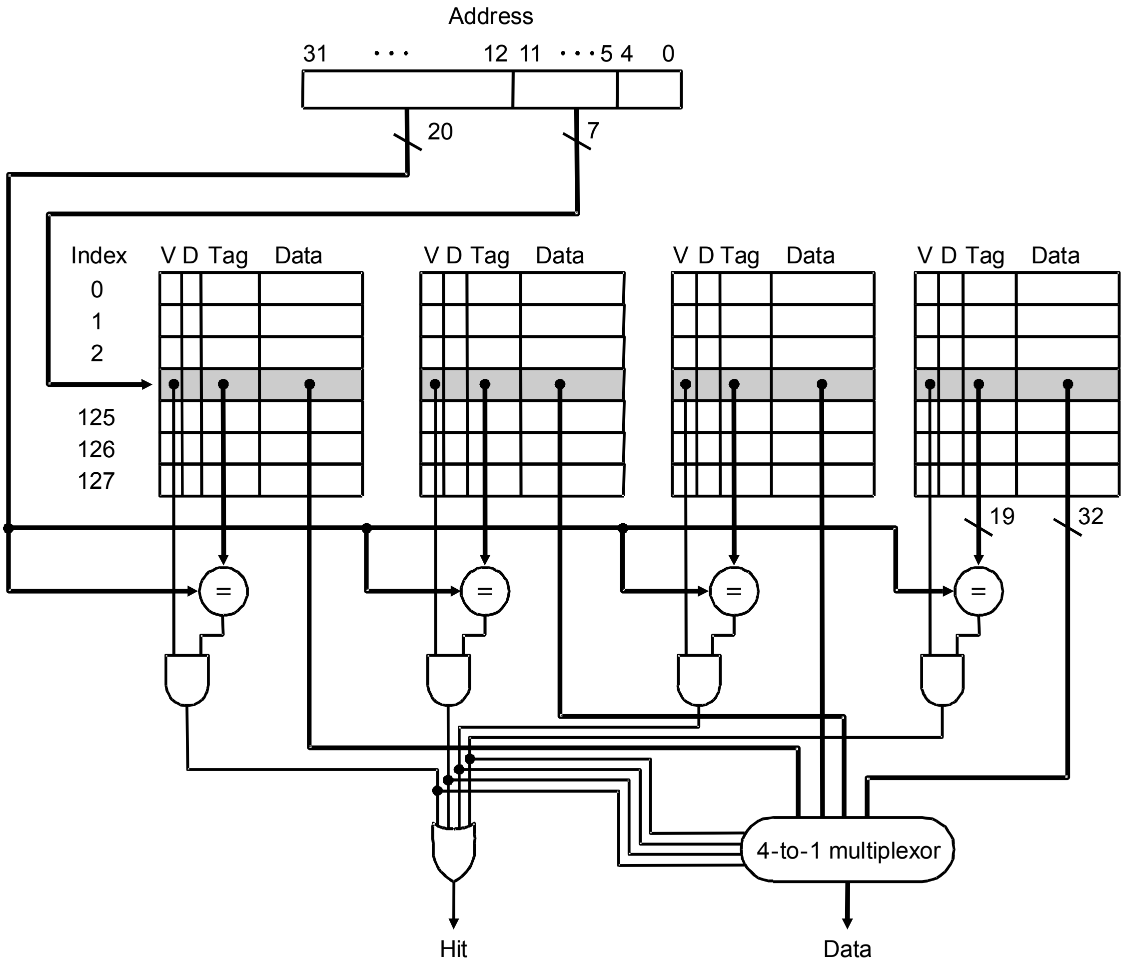


Figure 59 ***[Figure Caption]***

If most memory accesses are cache hit, MCU is able to get data immediately without wait states and the overall system performance will be higher. There are several factors that may affect cache hit rate:

- Cache size and the organization

  The larger the cache size is, the higher the hit rate will be. But the hit rate starts to saturate when cache size is larger than a threshold size. Normally the size of 16KB and above and two or four way can achieve a good hit rate.

- Program behavior

  If the system has several numbers of tasks and switches frequently between tasks, it may cause cache contents to be flushed out frequently. This is due to the fact that each time a new task runs cache will hold its data for a period of time for the opportunity of likely-be-used-again; however, the stored data might get flushed out before being used again if the following task requires the data that occupies the same cache entries. Interrupts can cause program flow to change dynamically and

reduces the benefit of cache. The interrupt handler code itself and the data it processes may cause cache to flush out data used by current task. Thus, after returning from interrupt handler to current task, the flushed data may need to be filled into cache again if it is required by the program routine. This causes performance degradation.

To assist software engineer tuning system performance, the cache controller in MT6516 records cache hit count and number of cacheable memory accesses. Cache hit rate can be obtained from dividing these two numbers.

The cache system also comprises a module called MPU (memory protection unit). MPU can prevent illegal memory access and specify which memory regions are cacheable or non-cacheable. Two fields in CACHE_CON register control the enable of MPU functions. MPU has its own registers to define memory region and associated regions. These settings only take effect after the enable bit in CACHE_CON is set to 1. For more details on the settings, please refer to MPU section of this design specification.

3.8.2 Cache Write Back

There are two different types of cache design to maintain the data consistency. One is cache write through and the other is cache write back. The second one can improve the performance especially when processors can generate writes as fast as or faster than the writes can be handled by the external memory; however, the implementation of write back is much harder than it of write through. When a cache line is dirty, four or eight words will be written back to external memory at once, and this will certainly occupy significant bus bandwidth. Therefore, decreases the overall efficiency. To solve this problem, a write buffer is necessary in the write back implementation. Once the writes get written into write buffer, processor can continue execution.

 © 2009 MediaTek Inc.

3.8.2.1 Write Back Implementation

When a cache hit happens on the write request, only the cache content gets modified and dirty bit gets set (in contrast to write through, write through modifies both the contents of cache and external memory). Now the content of the cache location where just got modified is inconsistent with that is in the external memory. When cache misses on the read request, line fill will be performed and a randomly selected cache line will be replaced, but before that, the dirty bits of that selected cache line have to be checked for the necessity of write back. If dirty bits are not set, line fill can proceed right away, and the selected cache line can be simplify flushed and replaced by a newly fetched line from external memory in which consists the requesting data. On the other hand, if one or both of the dirty bits are set, write back has to be performed before line fill. In that case, half or the whole cache line gets written into write buffer. To summarize above, the following figure is employed:

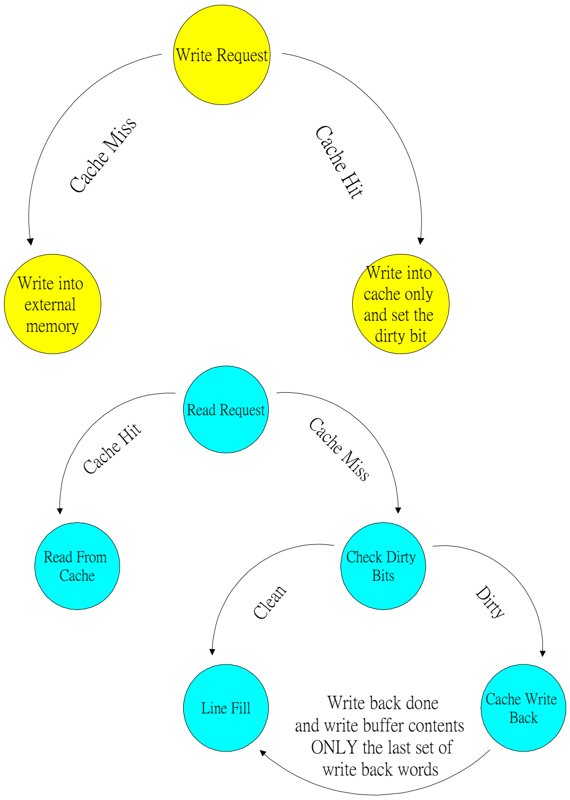


Figure 60 ***[Figure Caption]***

3.8.2.2 Write Buffer

Write buffer consists of address buffer, data buffer, buffer for HTRANS, buffer for HSIZE, buffer for HLOCK , and buffer for HBURST. The write buffer is first-in-first-out (FIFO) design, and depth of write buffer is eight

 © 2009 MediaTek Inc.

words. Since the outputs from code cache meet AMBA format, address buffer and data buffer are independent and the outputs of write buffer also meet AMBA format and are designed for pipelining.

3.8.3 Cache Operations

Upon power on, cache memory contains random values and these numbers are useless by MCU. MCU needs to flush out the cache content in each cache line before it gets utilized. Cache controller provides a register which, when written, could do operations on cache memory to fulfill the necessary prerequisite mentioned above. These are called cache operations, and the operations involve:

- Invalidate one cache line

  The user must give a memory address. If it is found within cache, that particular cache line contents the given address is invalidated. The invalidation is done by writing a zero to the valid bit at the corresponding tag line. Alternatively, the user can invalidate a cache line by specifying a set/way that maps to that cache line.

- Invalidate all cache lines

  The user needs not to specify an address. The cache controller clears valid bit in all tag lines when this operation is requested.

- Flush one cache line

  The user must give a memory address. If it is found within cache and the dirty bit or bits are set, that particular cache line contents the given address is flushed into write buffer. Alternatively, the user can invalidate a cache line by specifying a set/way that maps to that cache line.

- Flush all cache lines

  The user needs not to specify an address. The cache controller flushes all the cache lines with dirty bit or bits are set.

3.8.4 Cache Controller Register Definition

CACHE base address is assumed to be 0x5fff0000 (subject to change).

CACHE+00h Cache General Control Register CACHE_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							CACHESIZE						CNTEN1	CNTEN0	MPEN	MCEN
Type							RW						RW	RW	R/W	R/W
Reset							00						0	0	0	0

This register determines the size of cache; cache hit counter and the enable of MPU.

CACHESIZE Cache Size Select

- **00** no cache (112KB TCM)
- **01** 4KB, 1-way cache (108KB TCM)
- **10** 8KB, 2-way cache (104KB TCM)
- **11** 16KB, 4-way cache (96KB TCM)

CNTEN1 Enable cache hit counter 1

If enabled, cache controller will increment a 48-bit counter by one when a cache hit is detected. This number can provide a reference in performance evaluation for tuning the application programs. This counter increments only when the data is obtained from MPU cacheable region 4~7.

0 disable

1 enable

CNTEN0 Enable cache hit counter 0

If enabled, cache controller will increment a 48-bit counter by one when a cache hit is detected. This number can provide a reference in performance evaluation for tuning the application programs. This counter increments only when the data is obtained from MPU cacheable region 0~3.

0 disable

1 enable

MPEN Enable MPU comparison of read/write permission setting

If disabled, MCU can access any memory without restriction. If enabled, MPU will compare the address of MCU to MPU protection setting. If the MCU accessed address falls into restricted region, MPU will stop this memory access and send an “ABORT” signal to MCU. For details, please refer to MPU portion of the specification.

0 disable

1 enable

MCEN Enable MPU comparison of cacheable/non-cacheable setting

If disabled, MCU memory accesses are all non-cacheable, i.e., they will go through AHB bus (except for TCM access). If enabled, the setting in MPU will take effect. If MCU accesses a cacheable memory region, the cache controller will return the data in cache if it’s found in cache and will get the data through AHB bus only if a cache miss occurs. Please refer to MPU part of the specification for more details.

0 disable

1 enable

CACHE+04h Cache Operation CACHE_OP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TADDR[15:5]**											**OP[3:0]**				**EN**
Type	R/W											W				W1
Reset	0											0				0

This register defines the address and/or which kinds of cache operations to be taken. When MCU writes this register, the pipeline of MCU will be stopped for the cache controller to complete the operation. Bit 0 of the register must be written 1 to enable the command.

TADDR[31:5] Target Address

This field contains the address of invalidation operation. If OP[3:0]=0010, TADDR[31:5] is the address[31:5] of a memory whose line will be invalidated if it exists in the cache. If OP[3:0]=0100, TADDR[13:5] indicates the set, while TADDR[19:16] indicates which way to clear:

 © 2009 MediaTek Inc.

0001 way #0
0010 way #1
0100 way #2
1000 way #3

OP[3:0] Operation

This field determines which cache operations will be performed.

0001 invalidate all cache lines
0010 invalidate one cache line using address
0100 invalidate one cache line using set/way
1001 flush all cache lines
1010 flush one cache line using address
1100 flush one cache line using set/way

EN Enable command

This enable bit must be written 1 to enable the command.

1 enable
0 not enable

CACHE+08h Cache Hit Count 0 Lower Part CACHE_HCNT0L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CHIT_CNT0[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIT_CNT0[15:0]															
Type	R/W															
Reset	0															

CACHE+0Ch Cache Hit Count 0 Upper Part CACHE_HCNT0U

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIT_CNT0[47:32]															
Type	R/W															
Reset	0															

When CNTEN0 bit in CACHE_CON register is set to 1 (enabled), this register starts to record cache hit count until it is disabled. If the value increases to over maximum value (0xffffffffffff), it will be rolled over to 0 and continue counting. The 48 bit counter can provide a recording time of 31 days even if MCU runs at 104MHz and every cycle is a cache hit.

Note that before enabling the counter, it is recommended to write the initial value of zero to the counter.

CHIT_CNT0[47:0] Cache Hit Count 0

WRITE writing any value to CACHE_HCNT0L or CACHE_HCNT0U clears CHIT_CNT0 to all zeros
READ current counter value

CACHE+10h Cacheable Access Count 0 Lower Part — CACHE_CCNT0L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CACC_CNT0[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CACC_CNT0[15:0]															
Type	R/W															
Reset	0															

CACHE+14h Cacheable Access Count 0 Upper Part — CACHE_CCNT0U

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CACC_CNT0[47:32]															
Type	R/W															
Reset	0															

When CNTEN0 bit in CACHE_CON register is set to 1 (enabled), this register is incremented at each cacheable memory access (no matter it's a cache miss or a cache hit). If the value increases to over maximum value (0xffffffffffff), it will be rolled over to 0 and continue counting. For 104MHz MCU speed, if all memory accesses are cacheable and cache hit, this counter will overflow after (2^48) * 9.6ns = 31 days. This is the shortest time for the counter to overflow. In a more realistic case, the system will have cache misses, non-cacheable accesses, idle mode that makes the counter overflow at later time.

CACC_CNT0[47:0] Cache Access Count 0

WRITE writing any value to CACHE_CCNT0L or CACHE_CCNT0U clears CACC_CNT0 to all zeros
READ current counter value

The best way to use CACHE_HCNT0 and CACHE_CCNT0 is to set zero as initial value in both registers, enable both counters (set CNTEN0 to 1), run a portion of program to be benchmarked, stop the counters and get their values. Therefore during this period

$$Cache\ hit\ rate = \frac{CACHE_HCNT}{CACHE_CCNT} \times 100\% .$$

The cache hit rate value may help tune the performance of application program.

Note that CHIT_CNT0 and CACC_CNT0 only increment if the cacheable attribute is defined in MPU cacheable region 0~3.

CACHE+18h Cache Hit Count 1 Lower Part CACHE_HCNT1L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CHIT_CNT1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIT_CNT1[15:0]															
Type	R/W															
Reset	0															

CACHE+1Ch Cache Hit Count 1 Upper Part CACHE_HCNT1U

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIT_CNT1[47:32]															
Type	R/W															
Reset	0															

When CNTEN1 bit in CACHE_CON register is set to 1 (enabled), this register starts to record cache hit count until it is disabled. If the value increases to over maximum value (0xffffffffffff), it will be rolled over to 0 and continue counting. The 48 bit counter can provide a recording time of 31 days even if MCU runs at 104MHz and every cycle is a cache hit.

Note that before enabling the counter, it is recommended to write the initial value of zero to the counter.

CHIT_CNT1[47:0] Cache Hit Count

WRITE writing any value to CACHE_HCNT1L or CACHE_HCNT1U clears CHIT_CNT1 to all zeros

READ current counter value

CACHE+20h Cacheable Access Count 1 Lower Part CACHE_CCNT1L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CACC_CNT1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CACC_CNT1[15:0]															
Type	R/W															
Reset	0															

CACHE+24h Cacheable Access Count 1 Upper Part CACHE_CCNT1U

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CACC_CNT1[47:32]**															
Type	R/W															
Reset	0															

When CNTEN1 bit in CACHE_CON register is set to 1 (enabled), this register is incremented at each cacheable memory access (no matter it's a cache miss or a cache hit). If the value increases to over maximum value (0xffffffffffff), it will be rolled over to 0 and continue counting. For 104MHz MCU speed, if all memory accesses are cacheable and cache hit, this counter will overflow after (2^48) * 9.6ns = 31 days. This is the shortest time for the counter to overflow. In a more realistic case, the system will have cache misses, non-cacheable accesses, idle mode that makes the counter overflow at later time.

CACC_CNT1[47:0] Cache Access Count 1

WRITE writing any value to CACHE_CCNT1L or CACHE_CCNT1U clears CACC_CNT1 to all zeros

READ current counter value

The best way to use CACHE_HCNT1 and CACHE_CCNT1 is to set zero as initial value in both registers, enable both counters (set CNTEN1 to 1), run a portion of program to be benchmarked, stop the counters and get their values. Therefore during this period

$$Cache\ hit\ rate = \frac{CACHE_HCNT}{CACHE_CCNT} \times 100\% .$$

The cache hit rate value may help tune the performance of application program.

Note that CHIT_CNT1 and CACC_CNT1 only increment if the cacheable attribute is defined in MPU cacheable region 4~7.

3.9 MPU

3.9.1 General Description

The purpose of MPU is to provide a protection mechanism and cacheable indication of memory. The planned features of MPU include:

- 16-entry protection settings.

  Determine if MCU can read/write a memory region. If the setting doesn't allow MCU's particular access to a memory address, MPU will stop the memory access and issue "ABORT" signal to MCU, making it entering into "abort" mode. The exception handler must then process the situation.

- 16-entry cacheable settings.

  Determine a memory region is cacheable or not. If cacheable, MCU will keep a small copy in its cache after read accesses. If MCU requires the same data later, it can get it from the high-speed local copy, instead of from low-speed external memory.

Normally the protection and cacheable attributes are combined together for the same address range, as in the example of ARM946E. For greater flexibility, the MPU in MT6516 provides independent protection and cacheable settings. That is to say, the memory regions defined for memory protection and for cacheable are different and independent of each other.

The 4GB memory space is divided to 16 memory blocks with 256MB size each, i.e., MB0~MB15. EMI takes MB0~MB3, SYSRAM takes MB4, TCM uses MB5, APB peripherals MB8. The characteristics of these memory blocks are listed below:

- Read/write protection setting

  MB6 and above are always readable/writeable.
  MB0~MB4 and MB5 are determined by MPU.

- Cacheable setting

  MB4 and above are always non-cacheable.
  MB0~MB3 are determined by MPU.

3.9.2 Protection Settings

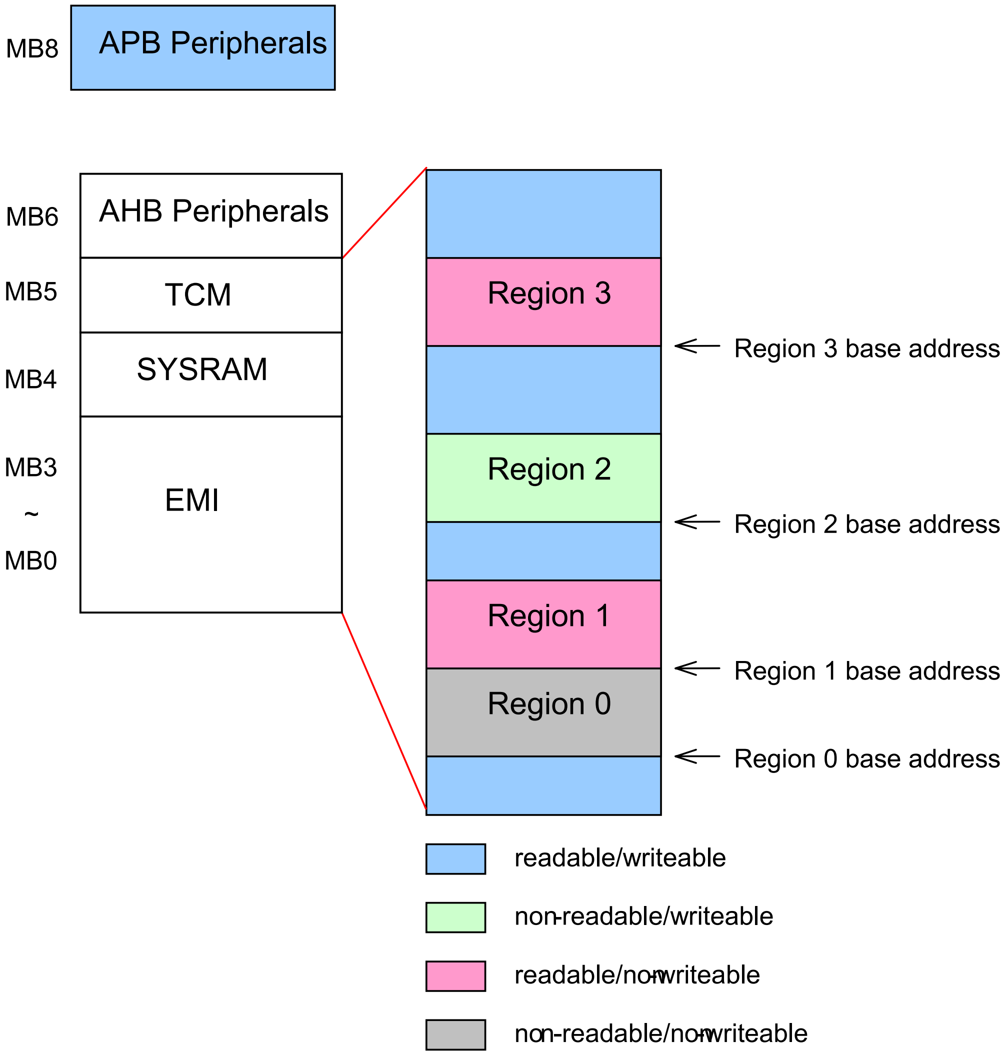


Figure 61 ***[Figure Caption]***

Figure 61 shows the protection setting in each memory block. Five regions are defined in the figure. Note that each region can be continuous or non-continuous to each other, and those address ranges not covered by any region are set to be readable/writeable automatically. One restriction exists: different regions must not overlap.

The user can define maximum 16 regions in MB0~MB4 and MB10. Each region has its own setting defined in a 32-bit register:

© 2009 MediaTek Inc.

31 ... 10	7	6 5	1	0
base address	00	prot	size	EN

- Region base address (22 bits)
- Region size (5 bits)
- Region protection attribute (2 bits)
- Enable bit (1 bit)

MPU will abort MCU if it accesses MB11~MB15 regions.

3.9.2.1 Region base address

Region base address defines the starting point of the memory region. The user needs only to specify several upper address bits. The number of valid address bits depends on the region size. The user must align the base address to a region-size boundary. For example, if a region size is 8KB, its base address must be a multiple of 8KB.

3.9.2.2 Region size

The bit representation of region size and its relationship with base address are listed as follows.

Region size	Bit encoding	Base address
1KB	00000	Bit [31:10] of region start address
2KB	00001	***Bit [31:11] of region start address***
4KB	***00010***	Bit [31:12] of region start address
8KB	00011	Bit [31:13] of region start address
16KB	00100	Bit [31:14] of region start address
32KB	00101	Bit [31:15] of region start address
64KB	00110	Bit [31:16] of region start address
128KB	00111	Bit [31:17] of region start address
256KB	01000	Bit [31:18] of region start address
512KB	01001	Bit [31:19] of region start address
1MB	01010	Bit [31:20] of region start address
2MB	01011	Bit [31:21] of region start address
4MB	01100	Bit [31:22] of region start address

Table 62 Region size and bit encoding

3.9.2.3 Region protection attribute

This attribute has two bits. The MSB determines read access permission, and the LSB for write access permission.

Bit encoding	Permission
00	non-readable / non-writeable

© 2009 MediaTek Inc.

10	***readable / non-writeable***
01	non-readable / writeable
11	readable / writeable

Table 63 Region protection attribute bit encoding

Note that bit encoding “11” allows full read/write permission, which is the case when no region is specified. So it is recommended to only specify regions with protection attribute “00”, “10” or “01”.

3.9.3 Cacheable Settings

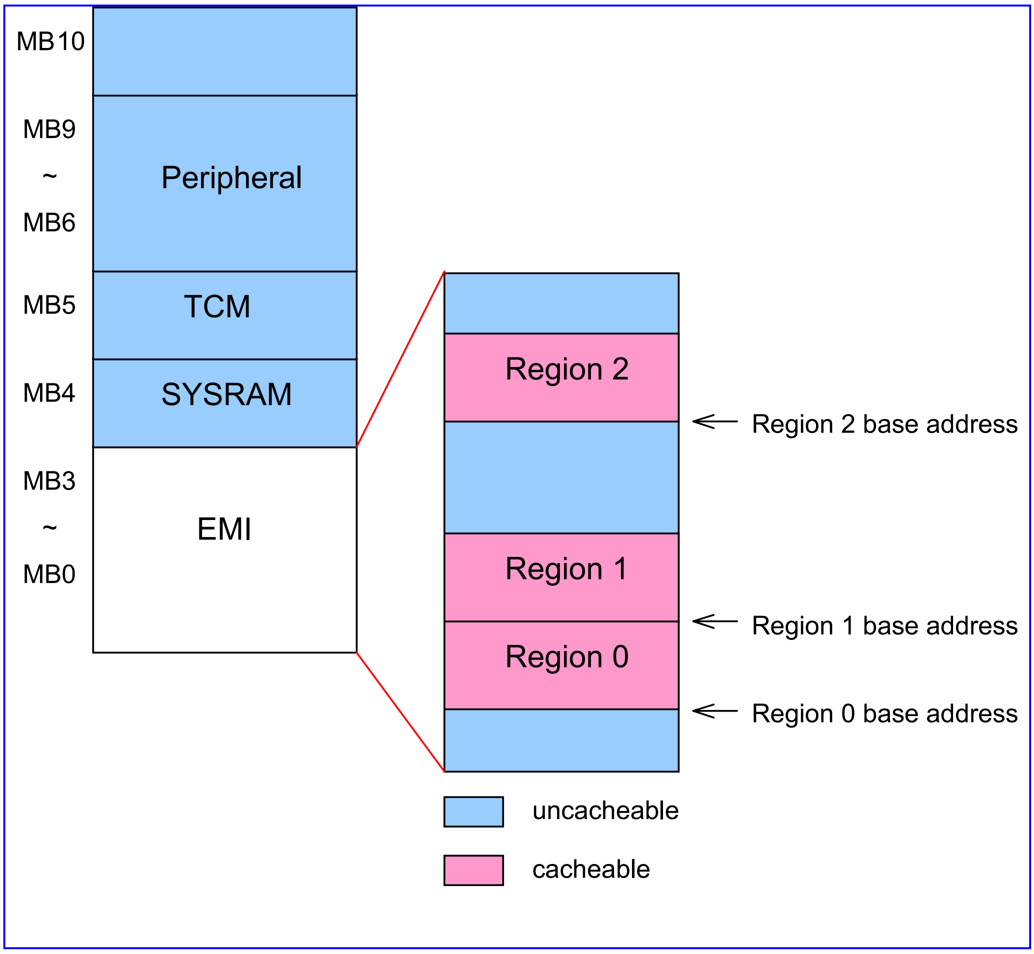


Figure 62 ***[Figure Caption]***

Cacheable setting and non-cacheable setting are similar to cache protection settings please refer to *Figure 7****. Note that each region can be continuous or non-continuous to each other. For those address ranges that not being covered by any region in the MPU cacheable settings are set to be uncacheable automatically. One restriction exists: different regions must not overlap.***

© 2009 MediaTek Inc.

The user can define maximum 16 regions in MB0~MB3. Each region has its own setting defined in a 32-bit register:

31 … 10	9 … 7	6	5 … 1	0
base address	000	C	size	EN

- Region base address (22 bits)
- Region size (5 bits)
- Region cacheable attribute (1 bit)
- Enable bit (1 bit)

The region base address and region size bit encoding are the same as those of protection setting. The user must also align the base address to a region-size boundary. The cacheable attribute has the following meaning.

Bit encoding	Attribute
0	uncacheable
1	cacheable

Table 64 Region cacheable attribute bit encoding

3.9.4 MPU Register Definition

MPU base address is assumed 0x5fff1000 (subject to change).

MPU+0000h Protection setting for region 0 MPU_PROT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]								ATTR[1:0]		SIZE[4:0]					E N
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 0.

BASEADDR Base address of this region

ATTR Protection attribute

- **00** non-readable / non-writeable
- **01** non-readable / writeable
- **10** readable / non-writeable
- 11 readable / writeable

SIZE size of this region

- **00000** 1KB
- **00001** 2KB
- **00010** 4KB

 © 2009 MediaTek Inc.

00011 8KB
00100 16KB
00101 32KB
00110 64KB
00111 128KB
01000 256KB
01001 512KB
01010 1MB
01011 2MB
01100 4MB

EN enable this region

0 Disable
1 Enable

MPU+0004h Protection setting for region 1 MPU_PROT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 1.

MPU+0008h Protection setting for region 2 MPU_PROT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 2.

MPU+000Ch Protection setting for region 3 MPU_PROT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

© 2009 MediaTek Inc.

This register sets protection attributes for region 3.

MPU+0010h Protection setting for region 4 MPU_PROT4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 4.

MPU+0014h Protection setting for region 5 MPU_PROT5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 5.

MPU+0018h Protection setting for region 6 MPU_PROT6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 6.

MPU+001Ch Protection setting for region 7 MPU_PROT7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**E N**
Type	R W								R W		R W					R W
Reset									11		00000					0

This register sets protection attributes for region 7.

MPU+0020h Protection setting for region 8 MPU_PROT8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 8.

MPU+0024h Protection setting for region 9 MPU_PROT9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 9.

MPU+0028h Protection setting for region 10 MPU_PROT10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 10.

MPU+002Ch Protection setting for region 11 MPU_PROT11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 11.

© 2009 MediaTek Inc.

MPU+0030h Protection setting for region 12 MPU_PROT12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 12.

MPU+0034h Protection setting for region 13 MPU_PROT13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 13.

MPU+0038h Protection setting for region 14 MPU_PROT14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 14.

MPU+003Ch Protection setting for region 15 MPU_PROT15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BASEADDR[15:10]**								**ATTR[1:0]**		**SIZE[4:0]**					**EN**
Type	RW								RW		RW					RW
Reset									11		00000					0

This register sets protection attributes for region 15.

 © 2009 MediaTek Inc.

MPU+0040h Cacheable setting for region 0 MPU_CACHE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 0.

BASEADDR Base address of this region

C Cacheable attribute

- **0** uncacheable
- **1** cacheable

SIZE size of this region

- **00000** 1KB
- **00001** 2KB
- **00010** 4KB
- **00011** 8KB
- **00100** 16KB
- **00101** 32KB
- **00110** 64KB
- **00111** 128KB
- **01000** 256KB
- **01001** 512KB
- **01010** 1MB
- **01011** 2MB
- **01100** 4MB

EN enable this region

- **0** Disable
- **1** Enable

MPU+0044h Cacheable setting for region 1 MPU_CACHE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 1.

MPU+0048h Cacheable setting for region 2 MPU_CACHE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 2.

MPU+004Ch Cacheable setting for region 3 MPU_CACHE3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 3.

MPU+0050h Cacheable setting for region 4 MPU_CACHE4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 4.

MPU+0054h Cacheable setting for region 5 MPU_CACHE5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 5.

MPU+0058h Cacheable setting for region 6 MPU_CACHE6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 6.

MPU+005Ch Cacheable setting for region 7 MPU_CACHE7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 7.

MPU+0060h Cacheable setting for region 8 MPU_CACHE8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 8.

MPU+0064h Cacheable setting for region 9 MPU_CACHE9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					EN
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 9.

MPU+0068h Cacheable setting for region 10 MPU_CACHE10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 10.

MPU+006Ch Cacheable setting for region 11 MPU_CACHE11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 11.

MPU+0070h Cacheable setting for region 12 MPU_CACHE12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 12.

MPU+0074h Cacheable setting for region 13 MPU_CACHE13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEADDR[31:16]															
Type	R W															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASEADDR[15:10]									C	SIZE[4:0]					E N
Type	R W									R W	R W					R W
Reset										0	00000					0

This register sets cacheable attributes for region 13.

© 2009 MediaTek Inc.

MPU+0078h Cacheable setting for region 14 MPU_CACHE14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**									**C**	**SIZE[4:0]**					**EN**
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 14.

MPU+007Ch Cacheable setting for region 15 MPU_CACHE15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASEADDR[31:16]**															
Type	RW															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASEADDR[15:10]**									**C**	**SIZE[4:0]**					**EN**
Type	RW									RW	RW					RW
Reset										0	00000					0

This register sets cacheable attributes for region 15.

3.10 Log Accelerator

3.10.1 General Description

In order to download more information from the network and to reduce the software loading, software only writes all information (the layer 1 and protocol service) at the specified external memory and triggers the hardware to transmit it though UART to the PC client.

Before transmitting data though UART, software needs to specify some settings. At first, the address of the layer 1 and PS data is needed. The amount of data is so huge that the data usually is located at the external memory. Software needs to specify the starting address and the size of the ring buffer. However, if the address is located at the internal ram (at the same layer of the AHB bus), it is acceptable. For MT6516, there is no the internal ram at the same layer with the log_acc. Therefore, the log_acc can not access the internal ram.

Besides the location of the original data, software needs to specify the target address. There are two ways to transmit data to UART.

i. The log_acc module writes all encoded data to the specified memory and then issues an interrupt to MCU after the whole block is transmitted. Then, software uses DMA to move the data to UART or USB.

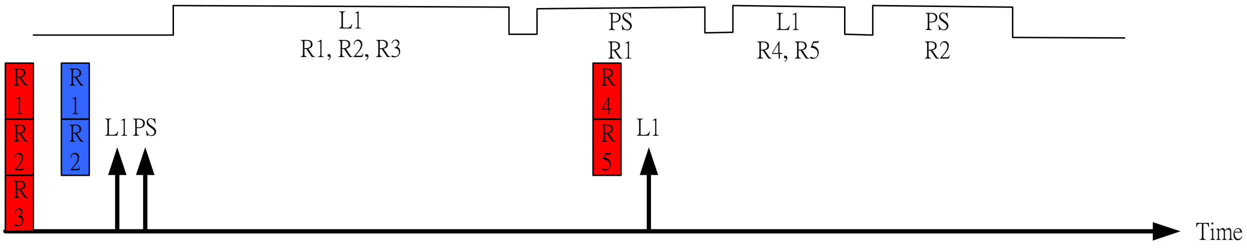
ii. The log_acc module writes the UART port directly.

In the first approach, software needs to set the specify memory for storing all encoded data. While ARM receives the IRQ signal from LOG_ACC, software can start to use DMA to move the data to UART or USB. This approach has a limit: if the encoded data is not word-aligned, SW should separate encoded data into two

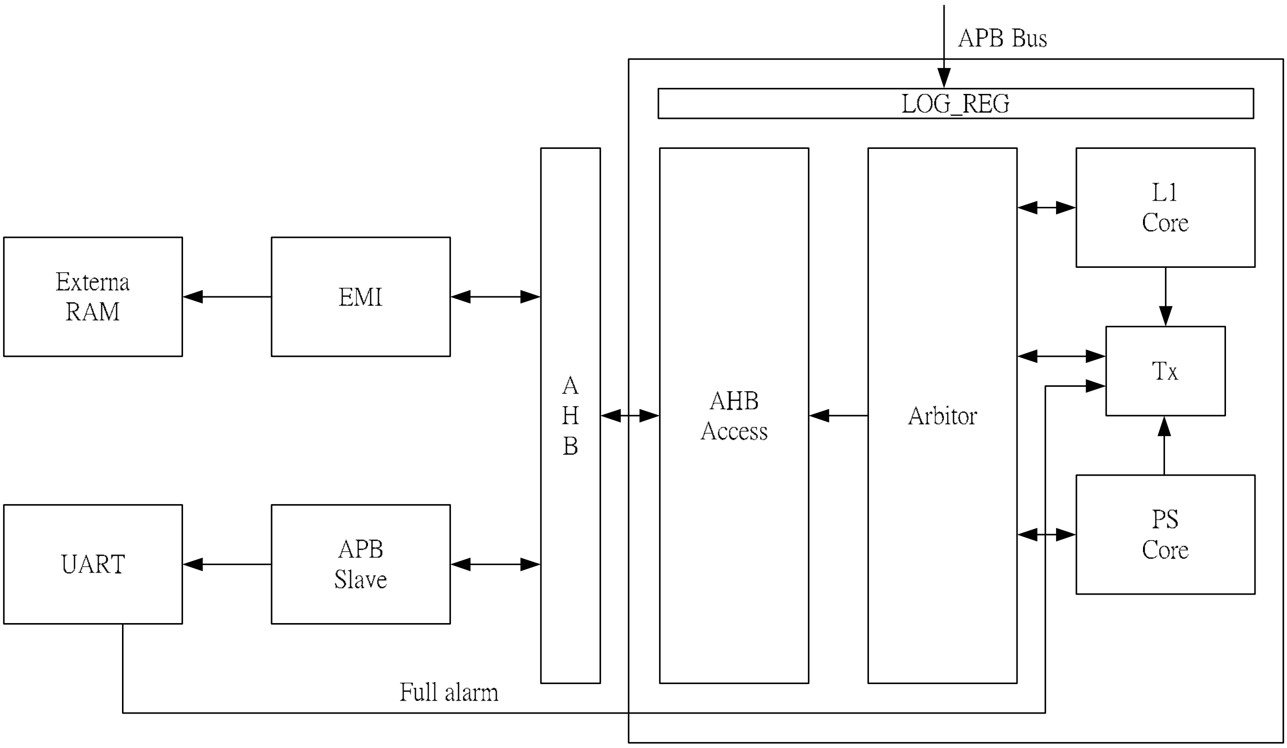
parts: one part is sent by “DMA size=word” way, and the remaining not-word-aligned data is sent by “DMA size=byte”. If SW uses “DMA size=byte” to send all data, the bus performance will be degraded and the transmission rate is too slow.

In the second approach, the LOG_ACC will write UART port directly by AHB single mode. Software needs to enable this function in the register the set the target address to UART port. SW should set the appreciate threshold value in “UARTn_FCR”, otherwise the log_acc will write too much data to UART FIFO. Although the log_acc can detect the status of the UART, the log_acc may not be the first master to access the UART while the UART FIFO is safe to send. If it happens (too much data), SW can try to use the DMA approach because this approach will hold the transmission until the FIFO is not full. If SW can grantee the one UART dedicated to the LOG_ACC module, then there is no such problem.

Log_acc Process Flow:



Red blocks belongs to the layer 1, and blue blocks belongs to the protocol stack. The layer 1 data has higher priority so that the layer 1 can send first if both layer 1 and PS data are ready. If there is no layer 1 data needed to send, then the log_acc starts to send PS data. However, if the layer 1 data is ready at the transmission of the PS data, the next transmission block will be the layer 1 data. After all layer 1data is sent, the next PS block continues to be sent.



In this HW version, HW supports to send data via two UARTs at the same time. While SW enables the dual master mode, the arbiter will become transparent and PS core will use the independent TX device and AHB access device. This mode setting must be set before the transmission. AHB Access1 & TX1 module are activated at the dual master mode.

3.10.2 Register Definition

Register Address	Register Function	Acronym
0x811b0000	Log Accelerator Control Setting	LOG_CON
0x811b0004	Layer 1 Base Address	LOG_LYR1_BASE
0x811b0008	Layer 1 Ring Buffer Size	LOG_LYR1_SIZE
0x811b000C	PS Base Address	LOG_PS_BASE
0x811b0010	PS Ring Buffer Size	LOG_PS_SIZE
0x811b0014	Target 0 Address (for Lyr1 or share mode)	LOG_TARGET0_ADDR
0x811b0018	Target 0 Size (for Lyr1 or share mode)	LOG_TARGET0_SIZE
0x811b001C	Target 0 Buffer Data Amount (for Lyr1 or share mode)	LOG_ TXBUF_CNT0
0x811b0020	Target 0 Log Status (for Lyr1 or share mode)	LOG_STATUS0
0x811b0024	UART_SEL for TARGET 0 (for Lyr1 and share mode)	LOG_UART_SEL0
0x811b0028	Layer 1 Pointer	LOG_LYR1_PT

0x811b002C	PS Pointer	LOG_PS_PT
0x811b0044	LOG IRQ STATUS	LOG_IRQ_STA
0x811b0050	TX_STATE	**LOG_DEBUG1**
0x811b0054	Write Pointer in TX module	**LOG_DEBUG2**
0x811b0058	Buffer Pointer in TX module	**LOG_DEBUG3**
0x811b005c	TX buffer count in TX module	**LOG_DEBUG4**
0x811b0060	Checksum in TX module	**LOG_DEBUG5**
0x811b0064	Encoded data to memory or UART	**LOG_DEBUG6**
0x811b0068	Escaping index & Encoded data in internal buffer	**LOG_DEBUG7**

Table 65 ***Log_acc Registers***

0x811b0000 Log Accelerator Control Setting LOG_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RST_B**	**LOG_DBG**			**TX1_UART**	**TX0_UART**	**TX1_MODE**	**TX0_MODE**				**AHBSP_MODE**		**DUAL_MASTER**	**PS_LOG_TRIG**	**LYR1_LOG_TRIG**
Type	R/W	R/W			R/W	R/W	R/W	R/W				R/W		R/W	W	W
Reset	1	0			0	0	0	0				0		0	0	0

LYR1_LOG_TRIG Trigger the hardware to check the ready byte of the monitored block. Once the hardware detects this trigger, then this bit will be desserted.

0 Disable

1 Trigger the hardware

PS_LOG_TRIG Trigger the hardware to check the ready byte of the monitored block. If the ready byte is no ok or the transmission is done, this bit will be disserted.

0 Disable

1 Trigger the hardware

TX0_MODE Approach 1 or 2 for TX0 module

0 Approach 1 (DMA -> UART or USB)

1 Approach 2 (UART directly), IRQ will not be issued while tx block is ready.

TX1_MODE Approach 1 or 2 for TX1 module (only valid at dual master mode)

0 Approach 1 (DMA -> UART or USB)

1 Approach 2 (UART directly), IRQ will not be issued while tx block is ready.

TX0_UART The specified UART will be dedicated to TX0 module, which is specified by LOG_UART_SEL0 register. Caution: 1. Do not change the setting at the transmission of LOG_ACC. 2. TX0_MODE should be set to 1.

0 UART can be shared to other modules

1 UART will be dedicated to LOG_ACC module only.

TX1_UART The specified UART will be dedicated to TX0 module, which is specified by LOG_UART_SEL0 register. Caution: 1. Do not change the setting at the transmission of LOG_ACC. 2. TX1_MODE should be set to 1.

0 UART can be shared to other modules

1 UART will be dedicated to LOG_ACC module only.

AHB SP_MODE Turn on AHB burst mode at the approach 2. (Caution: Under this special mode, the performance of the AHB increases, but this will violate the AHB protocol)

DUAL_MASTER Enable two AHB masters to access bus to support two UART devices transmission at the same time

0 Disable

1 Enable

LOG_DBG Select the output of debug information

0 Layer 1

1 Protocol Stack

RST_B Force the hardware to reset

0x811b0004 Layer 1 Base Address LOG_LYR1_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The address of the original data (4 word-aligned)

0x811b0008 Layer 1 Ring Buffer Size LOG_LYR1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									SIZE							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIZE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIZE The ring buffer size for the original data (4 word-aligned)

0x811b000C PS Base Address LOG_PS_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The address of the original data (4 word-aligned)

0x811b0010 PS Ring Buffer Size LOG_PS_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									SIZE							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIZE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIZE The ring buffer size for the original data (4 word-aligned)

0x811b0014 Target 0 Address (for Lyr1 or share mode) LOG_TARGET0_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The address of the encoded data (4 word-aligned, except for UART port) If your target is UART, please write the address of RX Buffer register in the UART.

0x811b0018 Target 0 Size (for Lyr1 or share mode) LOG_TARGET0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									SIZE							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIZE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The address of the encoded data (4 word-aligned)

0x811b001C Target 0 Buffer Data Amount (for Lyr1 or share mode) LOG_TXBUF_CNT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIZE The amount of the encoded data in the target buffer (Unit: byte)

0x811b0020 Target 0 Log Status (for Lyr1 or share mode) LOG_STATUS0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																RDY-TO-RD
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDY-TO-RD Indicate whether the target buffer is full. If the target is full or there is no block needed to process, the interrupt will be issued. After SW uses DMA move the data in the target buffer to the UART, SW should write 1'b0 to clear this register. The log_acc will wait until this bit is 1'b0.

0x811b0024 UART_SEL for TARGET 0 (for Lyr1 and share mode) LOG_UART_SEL0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														UART2	UART1	UART0
Type														R/W	R/W	R/W
Reset														0	0	0

UARTx Which UART is selected (Only valid when TX_MODE = 1). This setting is for indicating which UART's full alarm should be monitored.

0x811b0028 Layer 1 Pointer LOG_LYR1_PT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The pointer of the processing block (The real hardware address = Base_addr + Pointer)

0x811b002C PS Pointer LOG_PS_PT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR The pointer of the processing block (The real hardware address = Base_addr + Pointer)

© 2009 MediaTek Inc.

0x811b0044 LOG IRQ STATUS LOG_IRQ_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**IRQ1**	**IRQ0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQx Indicate which TX0/TX1 fire a interrupt. After reading this register, the register will be cleared.

- **0** No Interrupt
- **1** Interrupt

0x811b0050 LOG_ACC DEBUG Information (State machine) LOG_DEBUG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**CORE_STATE**			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**TX_STATE**												
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b0054 LOG_ACC Debug Information (Pointer) LOG_DEBUG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**WT_PT3**								**WT_PT2**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**WT_PT1**								**WT_PT0**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b0058 LOG_ACC Debug Information (Pointer) LOG_DEBUG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF_PT**		**BT_PT3**								**BT_PT2**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**BT_PT1**								**BT_PT0**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b005C LOG_ACC Debug Information (Counter) LOG_DEBUG4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**TX_CNT**														
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TOTAL_READ_DATA_COUNT**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b0060 LOG_ACC Debug Information (Check_Sum) LOG_DEBUG5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PS_CHECK_SUM								LYR1_CHECK_SUM							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b0064 LOG_ACC Debug Information (Encoded output data) LOG_DEBUG6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0x811b0068 LOG_ACC Debug Information (Buffer Data, Encoded index) LOG_DEBUG7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								INDEX	BUF_DATA							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								INDEX	BUF_DATA							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Use LOG_CON bit 21~16 to select which BUF_DATA should be read back. Use bit 22 to select MSB or LSB BUF_DATA to read back.

3.11 MD CONFIG Register

3.11.1 APB Bridge Register Map

REGISTER ADDRESS	REGISTER NAME	SYNONYM
8001_0000h	Hardware Version Register	HW_VER
8001_0004h	Software Version Register	SW_VER
8001_0008h	Hardware Code Register	HW_CODE
8001_0010h	Software Misc. Low Register	SW_MISC_L
8001_0014h	Software Misc. High Register	SW_MISC_H
8001_0020h	Hardware Misc. Register	HW_MISC
8001_0204h	Sleep Control Register	SLEEP_CON
8001_0208h	MCU Clock Control Register	MCUCLK_CON
8001_0300h	MD2GSYS Output Isolation Register	ISO_EN
8001_0304h	MD2GSYS Power Down Register	PWR_OFF
8001_0308h	MD2GSYS Memory Power Down Register	MEM_PDN
8001_030Ch	MD2GSYS Input Isolation Register	IN_ISO_EN
8001_0404h	APB Bus Control Register	APB_CON
8001_0408h	Security Boot Register	SECURITY_REG

8001_0500h	External Write Buffer Control Register	EXTB_SIZE
8001_0504h	VCXO_OFF Register	VCXO_OFF

3.11.2 Register Definitions

8001_0000h Hardware Version Register HW_VERSION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXTP**				**MAJREV**				**MINREV**							
Type	RO				RO				RO				RO			
Reset	7				A				0				0			

This register is used by software to determine the hardware version of the chip. The register contains a new value whenever each metal fix or major step is performed. All values are incremented by a step of 1.

MINREV Minor Revision of the chip

MAJREV Major Revision of the chip

EXTP This field shows the existence of Hardware Code Register that presents the Hardware ID while the value is other than zero.

8001_0004h Software Version Register SW_VERSION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXTP**				**MAJREV**				**MINREV**							
Type	RO				RO				RO				RO			
Reset	7				A				0				0			

This register is used by software to determine the software version used with this chip. All values are incremented by a step of 1.

MINREV Minor Revision of the software

MAJREV Major Revision of the software

EXTP This field shows the existence of Hardware Code Register that presents the Hardware ID when the value is other than zero.

8001_0008h Hardware Code Register HW_CODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CODE3**				**CODE2**				**CODE1**				**CODE0**			
Type	RO				RO				RO				RO			
Reset	6				5				1				6			

This register presents the Hardware ID. CODE1 & CODE0 can be programmed by efuse_dout[61:54].

8001_0010h Software Misc Low Register SW_MISC_L

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SW_MISC_L**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Spare registers for software control.

8001_0014h Software Misc High Register SW_MISC_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SW_MISC_H															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Spare registers for software control.

8001_0020h Hardware Misc Register HW_MISC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							IRDMA_LIMIT_CNT								IRDMA_LIMIT_EN	TDMA_IDN
Type							R/W								R/W	R/W
Reset							0								0	0

Spare registers for platform control.

TDMA_IDN TDMA Debug Unit on/off control

- **0** Disable
- **1** Enable.

IRDMA_LIMIT_EN Enable the limiter for IRDMA

- **0** Disable
- **1** Enable.

IRDMA_LIMIT_CNT Set the maximum count for limiter, IRDMA only issues the request when counting to the end of IRDMA_LIMIT_CNT.

8001_0204h Sleep Control Register SLEEP_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										AP_ACT_B			DSP2	DSP1	AHB	MDMCU
Type										R/W			R/W	R/W	WO	WO
Reset										~ICORE			1	1	0	0

MDMCU Stop the MCU Clock to force MCU Processor to enter sleep mode. MCU clock will be resumed as long as there is an interrupt request or system is reset.

- **0** MCU Clock is running
- **1** MCU Clock is stopped

AHB Stop the AHB Bus Clock to force the entire bus to enter sleep mode. AHB clock will be resumed as long as there is an interrupt request or system is reset.

- **0** AHB Bus Clock is running
- 2 AHB Bus Clock is stopped

DSP1 Stop the DSP1 Clock.

- **0** DSP Bus Clock is running
- **1** DSP Bus Clock is stopped

DSP2 Stop the DSP2 Clock.

0 DSP Bus Clock is running

1 DSP Bus Clock is stopped

AP_ACT_B Active AP MCU. After MD MCU finish the initialization of the memory and system setting for AP MCU. MD MCU can set the bit as "1" to activate AP MCU.

0 Enable AP MCU. (MD MCU clock stops)

1 Disable AP MCU.

8001_0208h MCU Clock Control Register MCUCLK_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ARM7_FSEL	MCU_FSEL			
Type												R/W	R/W			
Reset												0	7			

MCU_FSEL MCU clock frequency selection. This control register is used to control the output clock frequency of MCU Dynamic Clock Manager. The clock frequency is from 13MHz to 104MHz. The waveforms of the output clock are shown below. This register setting applies to the free-running clock, e.g. IRQ controller clock, in idle and slow idle mode. In normal mode this field is not working.

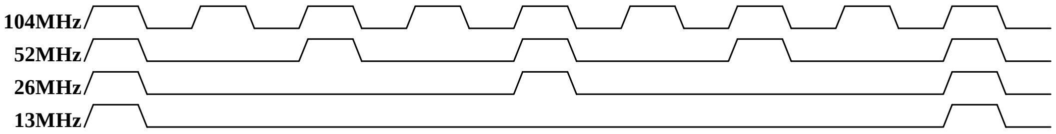


Figure 63 ***Output of Dynamic Clock Manager***

	High Speed Bus (and ARM7)	Low Speed Bus
0	13MHz	13MHz
1	26MHz	26MHz
2	Reserved	Reserved
3	52MHz	52MHz
4	Reserved	Reserved
5	Reserved	Reserved
6	Reserved	Reserved
7	104MHz	52MHz
Others	Reserved	

ARM7_FSEL ARM7 working clock frequency selection. This control register is used to control the ARM7 working clock frequency.

0 ARM7 clock runs at 52MHz

1 ARM7 clock runs at 104MHz

8001_0300h MD2GSYS Output Isolation Register ISO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																MD2G_ISO_EN
Type																R/W

Reset																0

MD2GSYS output isolation control

8001_0304h MD2GSYS Power Down Control Register PWR_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															MD2G_SW_PDN	MD2G_PDN
Type															R/W	R/W
Reset															1	0

MD2GSYS power down control register

MD2G_PDN MD2GSYS power down, this bit is valid in Software control mode.

- **0** Disable
- **1** Enable

MD2G_SW_PDN MD2GSYS HW/SW power down select

- **0** Hardware control
- **1** Software control

8001_0308h MD2GSYS Memory Power Down Register MEM_PDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															VFE_MEM_PDN	MD2G_MEM_PDN
Type															R/W	R/W
Reset															0	0

MD2GSYS memory power down control register

MD2G_MEM_PDN MD2GSYS memory power down

- **0** Disable
- **1** Enable

VFE_MEM_PDN VFE memory power down

- **0** Disable
- **1** Enable

8001_030Ch MD2GSYS Input Isolation Register IN_ISO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																MD2G_IN_ISO_EN
Type																R/W
Reset																0

MD2GSYS input isolation control register

8001_0404h APB Bus Control Register APB_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name		APBW 6	APBW 5	APBW 4	APBW 3	APBW 2	APBW 1	APBW 0		APBR6	APBR5	APBR4	APBR3	APBR2	APBR1	APBR 0
Type		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0		0	0	0	0	0	0	0

This register is used to control the timing of Read Cycle and Write Cycle on APB Bus.

APBR0-APBR6 Read Access Time on APB Bus

- **0** 1-Cycle Access
- **1** 2-Cycle Access

APBW0-APBW6 Write Access Time on APB Bus

- **0** 1-Cycle Access
- **1** 2-Cycle Access

8001_0408h Security Boot Register SECURITY_BOOT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SECURITY_BOOT							
Type									RO							
Reset									0							

8001_0500h MCU External Write Buffer Size Register WB_SIZE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																WB_SIZE
Type																R/W
Reset																0

This register is used to configure the external write buffer size of MCU.

WB_SIZE

- **0** 32-word data buffer and 8 address buffer
- **1** 16-word data buffer and 4 address buffer

8001_0504h VCXO_OFF Register VCXO_OFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																VCXO_OFF
Type																RO
Reset																vcxo_off

This register is used to monitor vcxo_off signal

3.12 MDMCUSYS CONFIG Register

In addition to the Pause Mode capability while in the Standby State, the software program can also put each peripheral independently into Power Down Mode while in the Active State by gating off their clock. The typical logic implementation is depicted in **Figure 13**. For all configuration bits, 1 signifies that the function is in Power Down Mode, and 0 means the function is in the Active Mode.

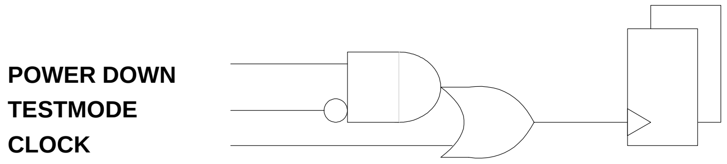


Figure 64 ***Power Down Control at Block Level***

REGISTER ADDRESS	REGISTER NAME	SYNONYM
811C_0000h	Clock Gating Control Status Register 0	MDMCUSYS_PDN_CON0
811C_0020h	Clock Gating Set Register 0	MDMCUSYS_PDN_SET0
811C_0040h	Clock Gating Clear Register 0	MDMCUSYS_PDN_CLR0
811C_0400h	Memory Delsel Control Register 0 (Used by Hardware)	MDMCUSYS_DELSEL0

Table 66 ***APB Bridge Register Map***

3.12.1 Register Definitions

811C_0000h Clock Gating Control Status Register 0 MDMCUSYS_PDN_CON0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**LA**														
Type		RO														
Reset		1														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**SIM2**	**SIM**					**AFC**			**TDMA**		**MDDMA**
Type					RO	RO					RO			RO		RO
Reset					1	1					1			1		1

MDMCU sub-system power down control status register (read only). Value 1 represents power down.

MDDMA Status of the Modem DMA Controller Power Down

TDMA Status of the TDMA Power Down

AFC Status of the AFC Power Down. This control is not being updated until both tdma_evtval and qbit_en are asserted.

SIM Status of the SIM Controller Power Down

SIM2 Status of the SIM2 Controller Power Down. SIM2 is actually power down when both AP and MD SIM2 power down bit are set.

LA Status of the Log Accelerator Power Down

811C_0020h Clock Gating Set Register 0 MDMCUSYS_PDN_SET0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name		**LA**														
Type		WO														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**SIM2**	**SIM**					**AFC**			**TDMA**		**MDDMA**
Type					WO	WO					WO			WO		WO

MDMCU sub-system power down set register, value 1 represents power down. For all registers addresses listed above, writing to the corresponding "SET" register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16'h0F0F, writing PDN_SET0 = 16'F0F0 will result in PDN_CON0 = 16'hFFFF.

811C_0040h Clock Gating Clear Register 0 MDMCUSYS_PDN_CLR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**LA**														
Type		WO														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**SIM2**	**SIM**					**AFC**			**TDMA**		**MDDMA**
Type					WO	WO					WO			WO		WO

MDMCU sub-system power down clear register, value 1 represents power up. For all registers addresses listed above, writing to the corresponding "Clear" register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_CON registers.
For example, if PDN_CON0 = 16'hFFFF, writing PDN_CLR0 = 16'F0F0 will result in PDN_CON0 = 16'h0F0F.

811C_0400h Memory Delsel Control Register 0 MDMCUSYS_DELSEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**MDSYSROM**				**TCM**		**L1CACHE**		**L1TAG**	
Type							R/W				R/W		R/W		R/W	
Reset							0101				11		11		10	

When writing bit 9, the effective value will be inversed. That is, Effect will be logic 1 when writing 0, and effect will be 0 when writing 1.

3.13 Reset Generation Unit (MDRGU)

Figure 35 shows the reset scheme used in MT6516. MT6516 provides three kinds of resets: hardware reset, watchdog reset, and software reset. MT6516 provides 8 resets which can be manual reset by individual RGU_USRSTx control registers.

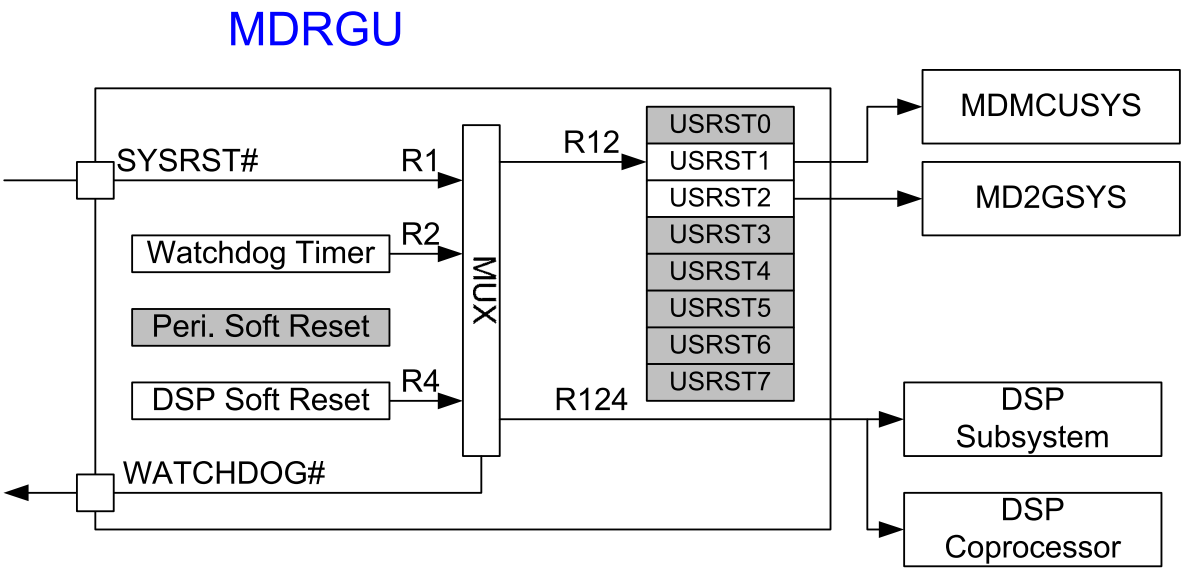


Figure 65 ***Reset Scheme Used in MT6516***

3.13.1 General Description

3.13.1.1 Hardware Reset

This reset is input through the SYSRST# pin, which is driven low during power-on. The hardware reset has a global effect on the chip: all digital and analog circuits are initialized, except the Real Time Clock module. The initial states of the MT6516 sub-blocks are as follows:.

- All analog circuits are turned off.
- All PLLs are turned off and bypassed. The 13 MHz system clock is the default time base.

3.13.1.2 Watchdog Reset

A watchdog reset is generated when the Watchdog Timer expires: the MCU software failed to re-program the timer counter in time. This situation is typically induced by abnormal software execution, which can be aborted by a hardwired watchdog reset. Hardware blocks that are affected by the watchdog reset are:

- MCU subsystem,
- DSP subsystem, and
- External components (trigged by software).

3.13.1.3 Software Resets

Software resets are local reset signals that initialize specific hardware components. For example, if hardware failures are detected, the MCU or DSP software may write to software reset trigger registers to reset those specific hardware modules to their initial states.

The following modules have software resets.

- DSP Core

 © 2009 MediaTek Inc.

- DSP Coprocessors

3.13.2 Register Definitions

RGU +0000h Watchdog Timer Control Register WDT_MODE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY[7:0]										WDTIN_DIS	AUTO-RESTART	IRQ	EXTEN	EXTPOL	ENABLE
Type	WO										R/W	R/W	R/W	R/W	R/W	R/W
Reset											1*	0	0	0	0	reset_en

ENABLE Enables the Watchdog Timer. The reset value depends on the ICORE: reset_en = ~ICORE.

- **0** Disables the Watchdog Timer.
- **1** Enables the Watchdog Timer.

EXTPOL Defines the polarity of the external watchdog pin.

- **0** Active low.
- 2 Active high.

EXTEN Specifies whether or not to generate an external watchdog reset signal.

- **0** The watchdog does not generate an external watchdog reset signal.
- **1** If the watchdog counter reaches zero, an external watchdog signal is generated.

IRQ Issues an interrupt instead of a Watchdog Timer reset. For debug purposes, RGU issues an interrupt to the MCU instead of resetting the system.

- **0** Disable.
- **1** Enable.

AUTO-RESTART Restarts the Watchdog Timer counter with the value of WDT_LENGTH while task ID is written into Software Debug Unit.

- **0** Disable. The counter restarts by writing KEY into the WDT_RESTART register.
- 2 Enable. The counter restarts by writing KEY into the WDT_RESTART register or by writing task ID into the software debug unit.

WDTIN_DIS If the other domain's watchdog affects the current domain's watchdog. ***WDTIN_DIS is only reset by external reset pin.**

- **0** This domain will be reset when other domain's watchdog is timeout.
- **1** This domain doesn't care about the other domain's watchdog.

KEY Write access is allowed if KEY=0x22.

RGU +0004h Watchdog Time-Out Interval Register WDT_LENGTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT[10:0]											KEY[4:0]				
Type	R/W											WO				
Reset	111_1111_1111b															

KEY Write access is allowed if KEY=08h.

TIMEOUT The counter is restarted with {TIMEOUT [10:0], 1_1111_1111b}. Thus the Watchdog Timer time-out period is a multiple of 512*T_{32k}=15.6ms.

 © 2009 MediaTek Inc.

RGU +0008h Watchdog Timer Restart Register WDT_RESTART

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[15:0]**															
Type	WO															
Reset																

KEY Restart the counter if KEY=1971h.

RGU +000Ch Watchdog Timer Status Register WDT_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WDT**	**SW_WDT**														
Type	RO	RO														
Reset	0	0														

WDT Indicates the cause of the watchdog reset.

- **0** Reset not due to Watchdog Timer.
- 2 Reset because the Watchdog Timer time-out period expired.

SW_WDT Indicates if the watchdog was triggered by software.

- **0** Reset not due to software-triggered Watchdog Timer.
- **1** Reset due to software-triggered Watchdog Timer.

NOTE: A system reset does not affect this register. This bit is cleared when the WDT_MODE register is written.

RGU +0010h CPU Peripheral Software Reset Register SW_PERIPH_RSTN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RESERVED**							
Type																
Reset																

RGU +0014h DSP Software Reset Register SW_DSP_RSTN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RST**								**KEY**							
Type	R/W															
Reset	0															

KEY Write access is allowed if KEY=0x48.

RST Controls the DSP System Reset Control.

- **0** No reset.
- **1** Invoke a reset.

RGU +0018h Watchdog Timer Reset Signal Duration Register WDT_RSTINTERVAL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**LENGTH[11:0]**											

Type					R/W											
Reset					FFFh											

LENGTH This register indicates the reset duration when Watchdog Timer times out. However, if the WDT_MODE register IRQ bit is set to 1, an interrupt is issued instead of a reset.

RGU+001Ch Watchdog Timer Software Reset Register WDT_SWRST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[15:0]**															
Type	WO															
Reset																

Software-triggered Watchdog Timer reset. If the register content matches the KEY, a watchdog reset is issued. However, if the WDT_MODE register IRQ bit is set to 1, an interrupt is issued instead of a reset.

KEY 1209h

RGU+0020h RGU user-defined reset 0 RGU_USRST0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST0[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0024h RGU user-defined reset 1 RGU_USRST1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST1[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0028h RGU user-defined reset 2 RGU_USRST2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST2[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+002ch RGU user-defined reset 3 RGU_USRST3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST3[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0030h RGU user-defined reset 4 RGU_USRST4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST4[7:0]**							
Type	WO								R/W							
Reset									0							

© 2009 MediaTek Inc.

RGU+0034h RGU user-defined reset 5 RGU_USRST5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST5[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+0038h RGU user-defined reset 6 RGU_USRST6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST6[7:0]**							
Type	WO								R/W							
Reset									0							

RGU+003ch RGU user-defined reset 7 RGU_USRST7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KEY[7:0]**								**USRST7[7:0]**							
Type	WO								R/W							
Reset									0							

KEY Write access is allowed if KEY=0xbb.

User-defined resets can trigger individual resets derived by MCU. When the USRSTx[7:0] is non-zero, the corresponding reset will be pull low. (All resets generated by RGU are active low)

USRSTx[6:0] is for a period of reset. It will decrease 1 per system clock when USRSTx[6:0] is not equal to 0. It is suitable for a predefined period reset.

USRSTx[7] is for a manual reset. It is only changed by MCU and suitable for a manual reset fully controlled by MCU.

Generally speaking, USRSTx[6:0] and USRSTx[7] will not be non-zero at the same time.

Example1:

0th cycle: USRSTx = 0x0. RESETx=1. MCU writes USRSTx = 0x3.

1st cycle: USRSTx = 0x3. RESETx=0.

2nd cycle: USRSTx = 0x2. RESETx=0.

3rd cycle: USRSTx = 0x1. RESETx=0.

4th cycle: USRSTx = 0x0. RESETx=1.

Example2:

0th cycle: USRSTx = 0x0. RESETx=1. MCU writes USRSTx = 0x80.

1st cycle: USRSTx = 0x80. RESETx=0.

2nd cycle: USRSTx = 0x80. RESETx=0.

~

N^{th} cycle: USRSTx = 0x80. RESETx=0. MCU writes USRSTx = 0x0.

$N+1^{th}$ cycle: USRSTx = 0x0. RESETx=1.

4 2.75G Modem Subsystem

4.1 Automatic Frequency Control (AFC) Unit

4.1.1 General description

The Automatic Frequency Control (AFC) unit provides the direct control of the oscillator for frequency offset and Doppler shift compensation. The block diagram is of the AFC unit depicted in **Figure 66.** The module utilizes a 13-bit D/A converter to achieve high-resolution control. Two modes of operation provide flexibility when controlling the oscillator; they are described as follows.

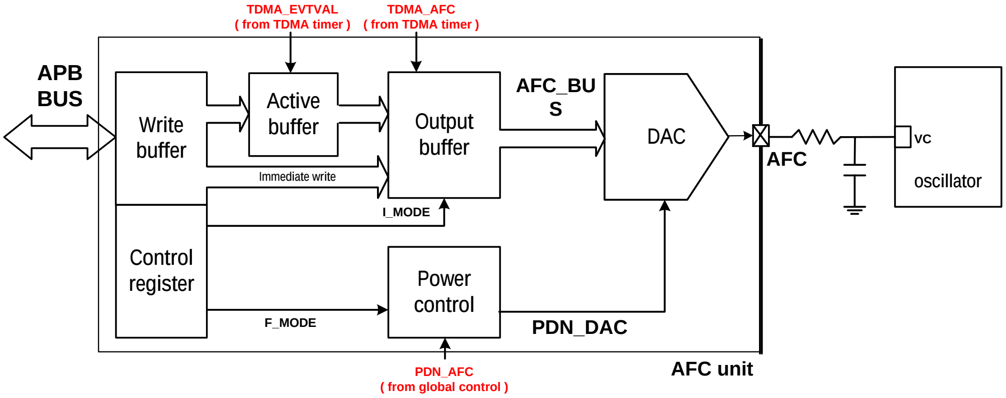


Figure 66 *The block diagram of the AFC controller*

In timer-triggered mode, the TDMA timer controls the AFC enabling events. Each TDMA frame can pulse at most four events. Double buffer architecture is supported. AFC values can be written to the write buffers. When the signal TDMA_EVTVAL is received, the values in the write buffers are latched into the active buffers. However, AFC values can also be written to the active buffers directly. Each event is associated with an active buffer sharing the same index. When a TDMA event is triggered by TDMA_AFC, the value in the corresponding active buffer takes effect. **Figure 67** shows a timing diagram of AFC events with respect to TX/RX/MX windows. In this mode, the D/A converter can stay powered on or be powered on for a programmable duration (256 quarter-bits, by default). The latter option is for power saving.

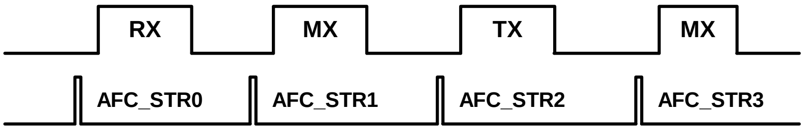


Figure 67 *Timing Diagram for the AFC Controller*

In immediate mode, the MCU can directly control the AFC value without event-triggering. The value written by the MCU takes effect immediately. In this mode, the D/A converter must be powered on continuously. When transitioning from immediate mode into timer-triggered mode (by setting flag I_MODE in the register AFC_CON to be 0), the D/A converter is kept powered on for a programmable duration (256 quarter-bits by default) if a TDMA_AFC is not been pulsed. The duration is prolonged upon receiving events.

External Devices turn on AFC DAC:

© 2009 MediaTek Inc.

Because the external devices (ex. Wifi & BlueTooth) need 26MHz Clock, AFC DAC is needed to power-up when baseband chip is under the sleep mode. The external devices can use SRCLKENAI to turn on AFC DAC and VAFC value will be as same as one that SW programs before the baseband chip enters the sleep mode.

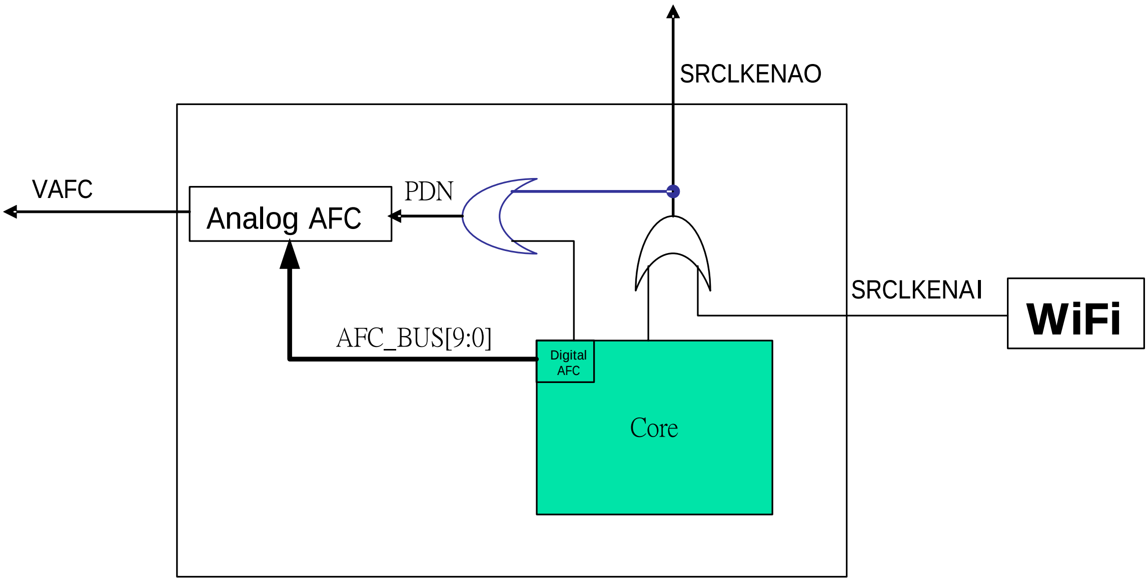


Figure 68 Architecture of external Devices turning on AFC DAC

While the Mobile or WiFi needs 26MHz, VAFC is needed to turn on. One difference of ON state is that VAFC is adjustable (ON mode) at the Mobile one turning on, but otherwise VAFC keeps at the same voltage (HOLD mode). In Figure 4, while WiFi is asserted (SRCLKENAI=1), VAFC will be turned on at the voltage which is set by the baseband chip before entering the sleep mode.

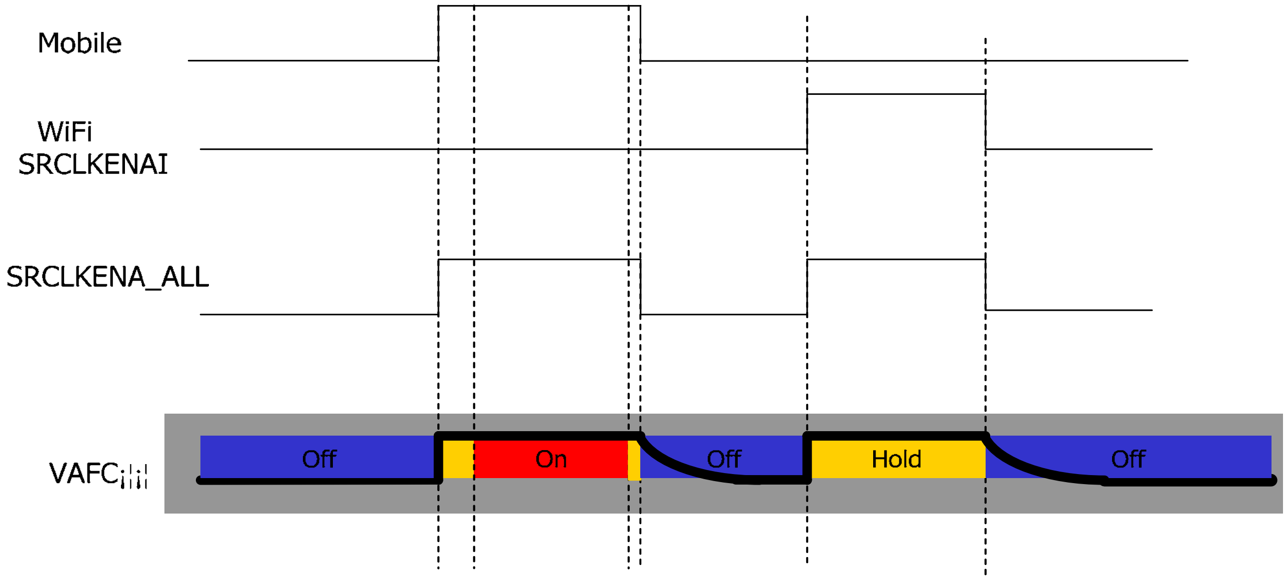


Figure 69 Waveform of AFC DAC ON/OFF

4.1.2 Register Definitions

Register Address	Register Function	Acronym
0x81200000	AFC control	**AFC_CON**
0x81200004	AFC control value 0	**AFC_DAT0**
0x81200008	AFC control value 1	**AFC_DAT1**
0x8120000C	AFC control value 2	**AFC_DAT2**
0x81200010	AFC control value 3	**AFC_DAT3**
0x81200014	AFC Power up period	**AFC_PUPER**

Table 67 ***AFC Registers***

0x81200000 AFC control register AFC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**ALY_ON**		**RDACT**	**F_MODE**	**FETENV**	**I_MODE**
Type											R/W		R/W	R/W	R/W	R/W
Reset											0		0	0	0	0

Four control modes are defined and can be controlled through the AFC control register. F_MODE enables the force power up mode. FETENV enables the direct write operation to the active buffer. I_MODE enables the immediate mode. RDACT enables the direct read operation from the active buffer. HOLD_ON enables the AFC DAC hold mode.

© 2009 MediaTek Inc.

I_MODE The flag enables the immediate mode. To enable the immediate mode also enable the force power up mode.

- **0** The immediate mode is not enabled.
- 1 The immediate mode is enabled.

FETENV The flag enables the direct write operation to the active buffer. Note the control flag is only applicable to the for data buffer including AFC_DAT0, AFC_DAT1, AFC_DAT2, and AFC_DAT3.

- **0** APB write to the write buffer.
- **1** APB write to the active buffer.

F_MODE The flag enables the force power up mode.

- **0** The force power up mode is not enabled.
- 1 The force power up mode is enabled.

RDACT The flag enables the direct read operation from the active buffer. Note the control flag is only applicable to the four data buffer including AFC_DAT0, AFC_DAT1, AFC_DAT2, and AFC_DAT3.

- **0** APB read from the write buffer.
- **1** APB read from the active buffer.

ALY_ON Force AFC DAC power on regardless of PDN_CONx setting in software power down control

- **0** Normal power down procedure of PDN_CONx
- **1** Force AFC DAC to keep power up state

While SRCLKENAI = 1'b1, AFC DAC will be turned on at the normal mode. SRCLKENAI is a gpio_mux pin. Make sure that the gpio mode is configured at the correct mode before BB enters the sleep mode.

0x81200004 AFC data register 0 AFC_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	15	4	3	2	1	0
Name				**AFCD**												
Type				R/W												

The register stores the AFC value for the event 0 triggered by the TDMA timer in timer-triggered mode. When the RDACT or FETENV bit (of the AFC_CON register) is set, the data transfer operates on the active buffer. When neither flag is set, the data transfer operates on the write buffer.

AFCD The AFC sample for the D/A converter.

Four registers (AFC_DAT0, AFC_DAT1, AFC_DAT2, AFC_DAT3) of the same type correspond to the event triggered by the TDMA timer. The four registers are summarized in **Table 1**.

Register Address	Register Function	Acronym
0x81200004	AFC control value 0	AFC_DAT0
0x81200008	AFC control value 1	AFC_DAT1
0x8120000C	AFC control value 2	AFC_DAT2
0x81200010	AFC control value 3	AFC_DAT3

Table 1 *AFC Data Registers*

Immediate mode can only use AFC_DAT0. In this mode, only the control value in the AFC_DAT0 write buffer is used to control the D/A converter. Unlike timer-triggered mode, the control value in AFC_DAT0 write buffer can bypass the active buffer stage and be directly coupled to the output buffer in immediate mode. To use

immediate mode, program the AFC_DAT0 in advance and then enable immediate mode by setting the I_MODE flag in the AFC_CON register.

The registers AFC_DATA0, AFC_DAT1, AFC_DAT2, and AFC_DAT3 have no initial values, thus the register must be programmed before any AFC event takes place. The AFC value for the D/A converter, i.e., the output buffer value, is initially 0 after power up before any event occurs.

0x81200014 AFC power up period AFC_PUPER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**PU_PER**												
Type				R/W												
Reset				ff												

This register stores the AFC power up period, which is 13 bits wide. The value ranges from 0 to 8191. If the I_MODE or F_MODE flag is set, this register has no effect since the D/A converter is powered up continuously. If neither flag is set, the register controls the power up duration of the D/A converter. During that period, the signal PDN_DAC in **Figure 66** is set to 1(power up).

PU_PER Stores the AFC power up period. After hardware power up, the field is initialized to 255.

4.1.3 Application Note

In GSM, DCS and PCS mobile system, GSM05.10 specify the MS carrier frequency should be accurate to within 0.1 ppm, or accurate to within 0.1 ppm compared to signals received from the BTS. In order to compensate the frequency error, which caused by BTS frequency error or Doppler shift effect, exists between MS and BTS, AFC (automatic frequency control) resides in MS should take the responsibility to adjust MS reference clock, 13MHz or 26MHz voltage controlled crystal oscillator (VCXO), in order to track up the frequency base of the signal from BTS.
Because AFC controls VCXO, VCXO characteristics will affect how to program AFC data. Typically, temperature compensated VCXO (TCVCXO) is used popularly in MS phone design, the following diagram illustrates the typical characteristic curve of TCVCXO in GSM900 and used on radio daughter board of MT62XX EVB. Therefore, by MT62XX EVB design, the tuning range 0~8191 of MT62XX 13bit DAC is mapping to the frequency error –8535~ 10208 Hz gotten from MT62XX report, and the slope is quite linear within 15% deviation in the full range of TCVCXO used for MT62XX DVB. By calculation, the slope is 2.367 Hz/step, which the important parameter used in AFC operation. Actually, the slope is only valid for GSM900 application, layer 1 will convert it for DCS1800 or PCS1900, automatically.

4.2 Automatic Power Control (APC) Unit

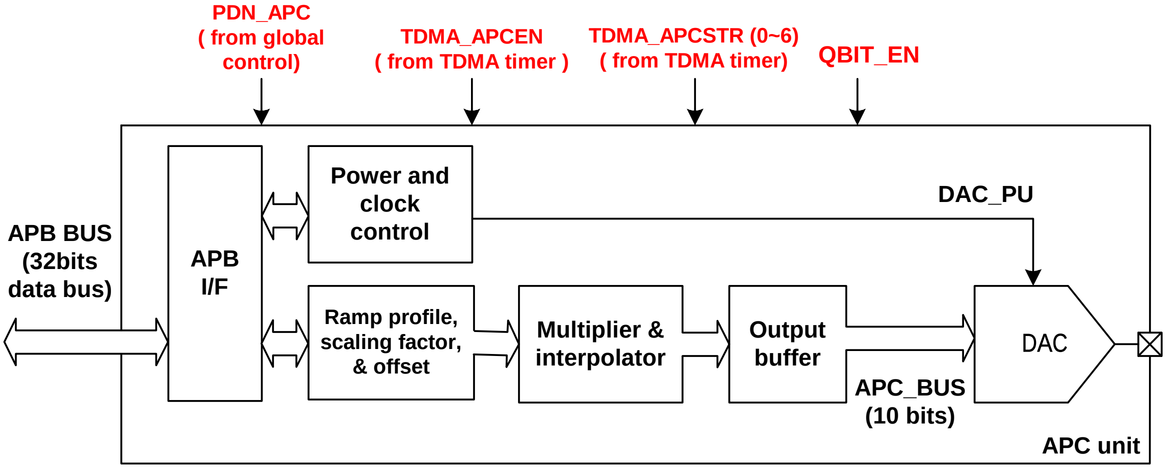
4.2.1 General Description

The Automatic Power Control (APC) unit controls the Power Amplifier (PA) module. Through APC unit, the proper transmit power level of the handset can be set to ensure that burst power ramping requirements are met. In one TDMA frame, up to 7 TDMA events can be enabled to support multi-slot transmission. In practice, 5 banks of ramp profiles are used in one frame to make up 4 consecutive transmission slots.
The shape and magnitude of the ramp profiles are configurable to fit ramp-up (ramp up from zero), intermediate ramp (ramp between transmission windows), and ramp-down (ramp down to zero) profiles.

Each bank of the ramp profile consists of 16 8-bit unsigned values, which are adjustable for different conditions.

The entries from one bank of the ramp profile are partitioned into two parts, with 8 values in each half. In normal operation, the entries in the left half are multiplied by a 10-bit left scaling factor, and the entries in the right half are multiplied by a 10-bit right scaling factor. The values are then truncated to form 16 10-bit intermediate values. Finally the intermediate ramp profile are linearly interpolated into 32 10-bit values and sequentially used to update the D/A converter. The block diagram of the APC unit is shown in **Figure 70 .**

The APB bus interface is 32 bits wide. Four write accesses are required to program each bank of ramp profile. The detailed register allocations are listed in **Table 68.**



•

Figure 70 *Block diagram of APC unit.*

4.2.2 Register Definitions

0X82040000 APC 1st ramp profile #0 APC_PFA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ENT3								ENT2							
Type	R/W								R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENT1								ENT0							
Type	R/W								R/W							

The register stores the first four entries of the first power ramp profile. The first entry resides in the least significant byte [7:0], the second entry in the second byte [15:8], the third entry in the third byte [23:16], and the fourth in the most significant byte [31:24]. Since this register provides no hardware reset, the programmer must configure it before any APC event takes place.

ENT3 The field signifies the 4th entry of the 1st ramp profile.

ENT2 The field signifies the 3rd entry of the 1st ramp profile.

ENT1 The field signifies the 2nd entry of the 1st ramp profile.

ENT0 The field signifies the 1st entry of the 1st ramp profile.

 © 2009 MediaTek Inc.

The overall ramp profile register definition is listed in **Table 68.**

Register Address	Register Function	Acronym
0x82040000	APC 1st ramp profile #0	APC_PFA0
0x82040004	APC 1st ramp profile #1	APC_PFA1
0x82040008	APC 1st ramp profile #2	APC_PFA2
0x8204000C	APC 1st ramp profile #3	APC_PFA3
0x82040020	APC 2nd ramp profile #0	APC_PFB0
0x82040024	APC 2nd ramp profile #1	APC_PFB1
0x82040028	APC 2nd ramp profile #2	APC_PFB2
0x8204002C	APC 2nd ramp profile #3	APC_PFB3
0x82040040	APC 3rd ramp profile #0	APC_PFC0
0x82040044	APC 3rd ramp profile #1	APC_PFC1
0x82040048	APC 3rd ramp profile #2	APC_PFC2
0x8204004C	APC 3rd ramp profile #3	APC_PFC3
0x82040060	APC 4th ramp profile #0	APC_PFD0
0x82040064	APC 4th ramp profile #1	APC_PFD1
0x82040068	APC 4th ramp profile #2	APC_PFD2
0x8204006C	APC 4th ramp profile #3	APC_PFD3
0x82040080	APC 5th ramp profile #0	APC_PFE0
0x82040084	APC 5th ramp profile #1	APC_PFE1
0x82040088	APC 5th ramp profile #2	APC_PFE2
0x8204008C	APC 5th ramp profile #3	APC_PFE3
0x820400A0	APC 6th ramp profile #0	APC_PFF0
0x820400A4	APC 6th ramp profile #1	APC_PFF1
0x820400A8	APC 6th ramp profile #2	APC_PFF2
0x820400AC	APC 6th ramp profile #3	APC_PFF3
0x820400C0	APC 7th ramp profile #0	APC_PFG0
0x820400C4	APC 7th ramp profile #1	APC_PFG1
0x820400C8	APC 7th ramp profile #2	APC_PFG2
0x820400CC	APC 7th ramp profile #3	APC_PFG3

***Table 68** APC ramp profile registers*

0x82040010 APC 1st ramp profile left scaling factor APC_SCAL0L

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**SF**									
Type							R/W									
Reset							1_0000_0000									

The register stores the left scaling factor of the 1st ramp profile. This factor multiplies the first 8 entries of the 1st ramp profile to provide the scaled profile, which is then interpolated to control the D/A converter.

After a hardware reset, the initial value of the register is 256. In this case, no scaling is done (each entry of the ramp profile is multiplied by 1), because the 8 least significant bits are truncated after multiplication.

The overall scaling factor register definition is listed in **Table 69 .**

SF Scaling factor. After a hardware reset, the value is 256.

0x82040014 APC 1st ramp profile right scaling factor APC_SCAL0R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**SF**									
Type							R/W									
Reset							1_0000_0000									

The register stores the right scaling factor of the 1st ramp profile. This factor multiplies the last 8 entries of the 1st ramp profile to provide the scaled profile, which is then interpolated to control the D/A converter.

After a hardware reset, the initial value of the register is 256. In this case, no scaling is done (each entry of the ramp profile is multiplied by 1), because the 8 least significant bits are truncated after multiplication.

The overall scaling factor register definition is listed in **Table 69 .**

SF Scaling factor. After a hardware reset, the value is 256.

0x82040018 APC 1st ramp profile offset value APC_OFFSET0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**OFFSET**									
Type							R/W									
Reset							0									

There are 7 offset values for the corresponding ramp profile.

The 1st offset value also serves as the pedestal value. The value is used to power up the APC D/A converter before the RF signals start to transmit. The D/A converter is then biased on the value, to provide the initial control voltage for the external control loop. The exact value depends on the characteristics of the external components. The timing to output the pedestal value is configurable through the TDMA_BULCON2 register of the timing generator; its valid range is 0~127 quarter-bits of time after the baseband D/A converter is powered up.

OFFSET Offset value for the corresponding ramp profile. After a hardware reset, the default value is 0. The overall offset register definition is listed in **Table 69.**

Register Address	Register Function	Acronym
0x82040010	APC 1st ramp profile left scaling factor	APC_SCAL0L
0x82040014	APC 1st ramp profile right scaling factor	APC_SCAL0R
0x82040018	APC 1st ramp profile offset value	APC_OFFSET0
0x82040030	APC 2nd ramp profile left scaling factor	APC_SCAL1L
0x82040034	APC 2nd ramp profile right scaling factor	APC_SCAL1R
0x82040038	APC 2nd ramp profile offset value	APC_OFFSET1

0x82040050	APC 3rd ramp profile left scaling factor	APC_SCAL2L
0x82040054	APC 3rd ramp profile right scaling factor	APC_SCAL2R
0x82040058	APC 3rd ramp profile offset value	APC_OFFSET2
0x82040070	APC 4th ramp profile left scaling factor	APC_SCAL3L
0x82040074	APC 4th ramp profile right scaling factor	APC_SCAL3R
0x82040078	APC 4th ramp profile offset value	APC_OFFSET3
0x82040090	APC 5th ramp profile left scaling factor	APC_SCAL4L
0x82040094	APC 5th ramp profile right scaling factor	APC_SCAL4R
0x82040098	APC 5th ramp profile offset value	APC_OFFSET4
0x820400B0	APC 6th ramp profile left scaling factor	APC_SCAL5L
0x820400B4	APC 6th ramp profile right scaling factor	APC_SCAL5R
0x820400B8	APC 6th ramp profile offset value	APC_OFFSET5
0x820400D0	APC 7th ramp profile left scaling factor	APC_SCAL6L
0x820400D4	APC 7th ramp profile right scaling factor	APC_SCAL6R
0x820400D8	APC 7th ramp profile offset value	APC_OFFSET6

***Table 69** APC scaling factor and offset value registers*

0x820400E0h APC control register APC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**GSM**	**FPU**
Type															R/W	R/W
Reset															1	0

GSM Defines the operation mode of the APC module. In GSM mode, each frame has only one slot, thus only one scaling factor and one offset value must be configured. If the GSM bit is set, the programmer needs only to configure APC_SCAL0L and APC_OFFSET0. If the bit is not set, the APC module is operating in GPRS mode.

- **0** The APC module is operating in GPRS mode.
- **1** The APC module is operating in GSM mode. Default value.

FPU Forces the APC D/A converter to power up. Test only.

- **0** The APC D/A converter is not forced to power up. The converter is only powered on when the transmission window is opened. Default value.
- **1** The APC D/A converter is forced to power up.

4.2.3 Ramp Profile Programming

The first value of the first normalized ramp profile must be written in the least significant byte of the APC_PFA0 register. The second value must be written in the second least significant byte of the APC_PFA0, and so on.

Each ramp profile can be programmed to form an arbitrary shape.

The start of ramping is triggered by one of the TDMA_APCSTR signals. The timing relationship between TDMA_APCSTR and TDMA slots is depicted in **Figure 71** for 4 consecutive time slots case. The power

ramping profile must comply with the timing mask defined in GSM SPEC 05.05. The timing offset values for 7 ramp profiles are stored in the TDMA timer register from TDMA_APC0 to TDMA_APC6.

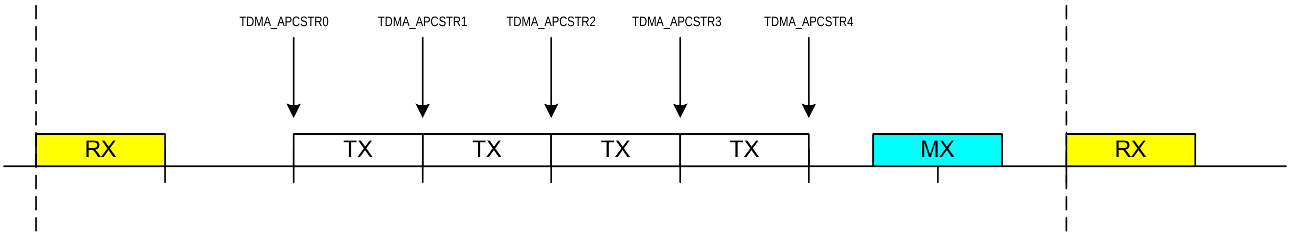


Figure 71 *Timing diagram of TDMA_APCSTR.*

Because the APC unit provides more than 5 ramp profiles, up to 4 consecutive transmission slots can accommodated. The 2 additional ramp profiles are useful particularly when the timing between the last 2 transmission time slots and CTIRQ is uncertain; software can begin writing the ramp profiles for the succeeding frame during the current frame, alleviating the risk of not writing the succeeding frame's profile data in time.

In GPRS mode, to fit the intermediate ramp profile between different power levels, a simple scaling scheme is used to synthesize the ramp profile. The equation is as follows:

$$\mathrm{DA}_0 = \mathrm{OFF} + \mathrm{S}_0 \cdot \frac{\mathrm{DN}_{15,pre} + \mathrm{DN}_0}{2}$$

$$\mathrm{DA}_{2k} = \mathrm{OFF} + \mathrm{S}_l \cdot \frac{\mathrm{DN}_{k-1} + \mathrm{DN}_k}{2}, k = 1,\ldots,15$$

$$\mathrm{DA}_{2k+1} = \mathrm{OFF} + \mathrm{S}_l \cdot \mathrm{DN}_k, k = 0,1,\ldots,15$$

$$l = \begin{cases} 0, & \text{if } 8 > k \geq 0 \\ 1, & \text{if } 15 \geq k \geq 8 \end{cases}$$

where **DA** = the data to present to the D/A converter,
DN = the normalized data which is stored in the register **APC_PF*n***,
S_0 = the left scaling factor stored in register **APC_SCAL*n*L**,
S_1 = the right scaling factor stored in register **APC_SCAL*n*R**, and
OFF = the offset value stored in the register **APC_OFFSET*n***.

The subscript ***n*** denotes the index of the ramp profile.

The ramp calculation before interpolation is as depicted in **Figure 72**.

During each ramp process, each word of the normalized profile is first multiplied by 10-bit scaling factors and added to an offset value to form a bank of 18-bit words. The first 8 words (in the left half part as in **Figure 72**) are multiplied by the left scaling factor **S_0** and the last 8 words (in the right half part as in **Figure 72**) are multiplied by the right scaling factor **S_1**. The lowest 8 bits of each word are then truncated to get a 10-bit result. The scaling factor is 0x100, which represents no scaling on reset. A value smaller than 0x100 scales the ramp profile down, and a value larger than 100 scales the ramp profile up.

 © 2009 MediaTek Inc.

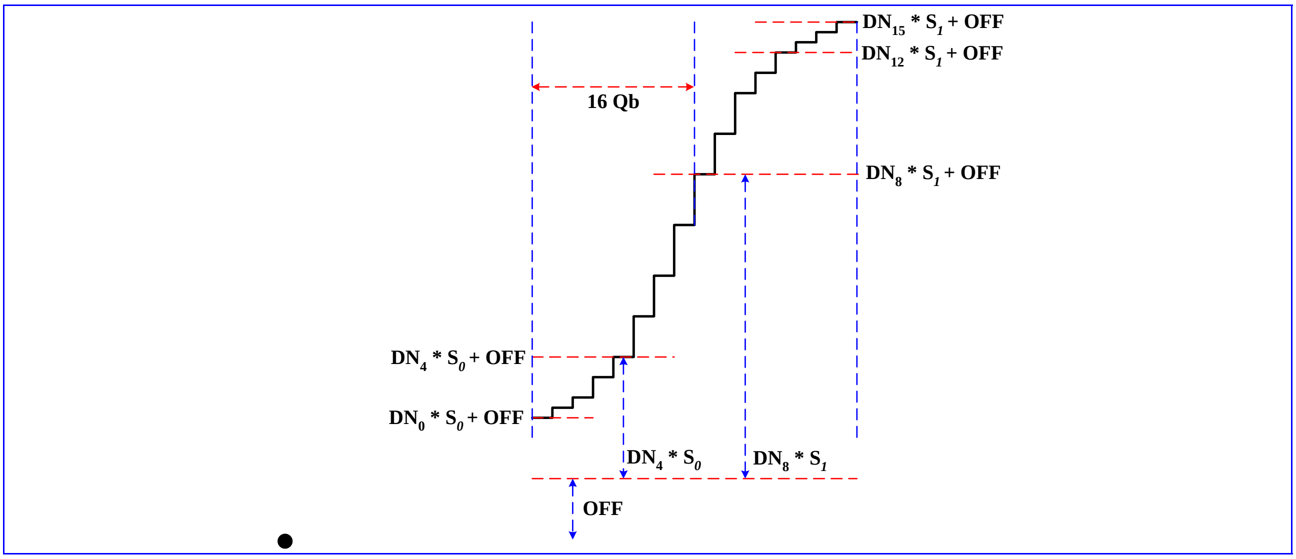


Figure 72 *The timing diagram of the APC ramp.*

The 16 10-bit words are linearly interpolated into 32 10-bit words. A 10-bit D/A converter is then used to convert these 32 ramp values at a rate of 1.0833 MHz, that is, at quarter-bit rate. The timing diagram is shown in **Figure 73** and the final value is retained on the output until the next event occurs.

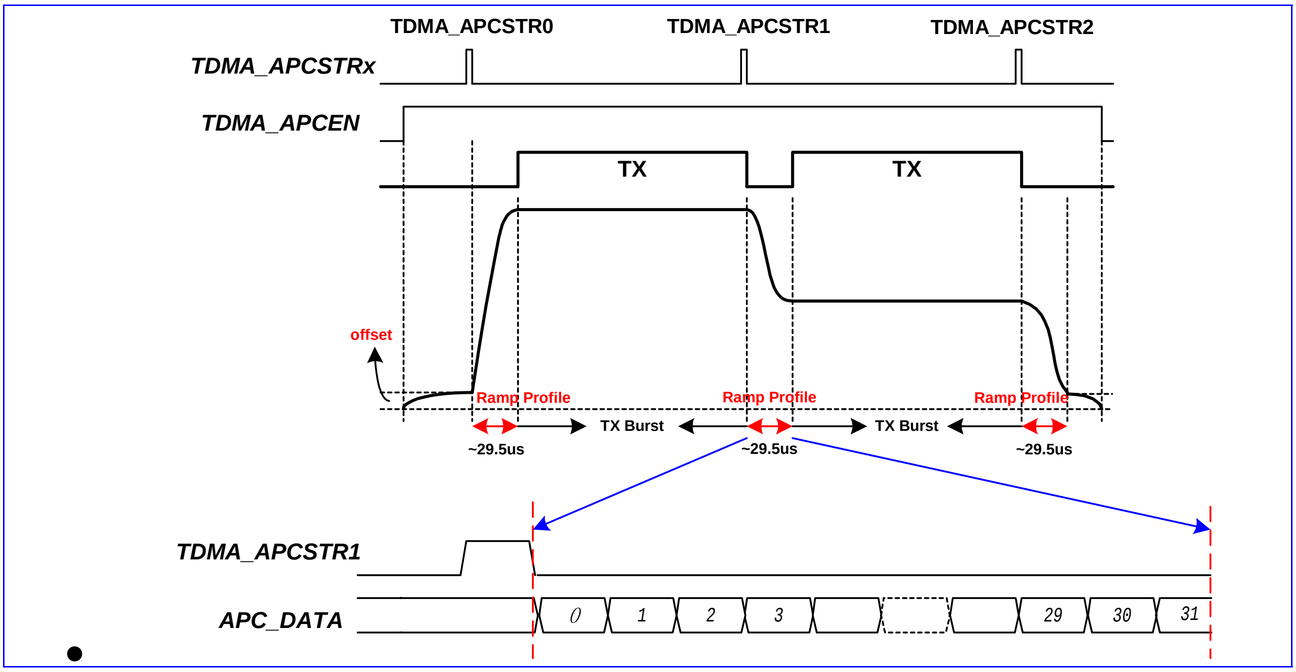


Figure 73 *Timing diagram of the APC ramping.*

© 2009 MediaTek Inc.

The APC unit is only powered up when the APC window is open. The APC window is controlled by configuring the TDMA registers TDMA_BULCON1 and TDMA_BULCON2. Please refer to the TDMA timer unit for more detailed information.

The first offset value stored in the register APC_OFFSET0 also serves as the pedestal value, which is used to provide the initial power level for the PA.

Since the profile is not double-buffered, the timing to write the ramping profile is critical. The programmer must be restricted from writing to the data buffer during the ramping process, otherwise the ramp profile may be incorrect and lead to a malfunction.

4.3 Baseband Front End

Baseband Front End is a modem interface between TX/RX mixed-signal modules and digital signal processor (DSP). We can divide this block into two parts (see **Figure 74**). The first is the uplink (transmitting) path, which converts bit-stream from DSP into digital in-phase (I) and quadrature (Q) signals for TX mixed-signal module. The second part is the downlink (receiving) path, which receives digital in-phase (I) and quadrature (Q) signals from RX mixed-signal module, performs FIR filtering and then sends results to DSP. **Figure 74** illustrates interconnection around Baseband Front End. In the figure the shadowed blocks compose Baseband Front End.

To enhance the capability of data processing of mobile phone and base station, the Enhanced Data for GSM Evolution (EDGE), which used 8PSK Modulation rather than GMSK Modulation in GSM system may provide the triple data transmission rate of 384 kbps for system to supply the solution of voice, data, Internet linkage, and other kinds of mutual linkage, while 3bits per symbols in 8PSK Modulation and 1 bit per symbol in GMSK Modulation.

The uplink path is mainly composed of GMSK Modulator or 8PSK Modulator and uplink parts of Baseband Serial Ports, and the downlink path is mainly composed of RX digital FIR filter and downlink parts of Baseband Serial Ports. Baseband Serial Ports is a serial interface used to communicate with DSP. In addition, there is a set of control registers in Baseband Front End that is intended for control of TX/RX mixed-signal modules, inclusive of several compensation circuit :calibration of I/Q DC offset, I/Q Quadrature Phase Compensation and I/Q Gain Mismatch of uplink analog-to-digital (D/A) converters as well as I/Q Gain Mismatch for downlink digital-to-analog (A/D) converters in TX/RX mixed-signal modules. The timing of bit streaming through Baseband Front End is completely under control of TDMA timer. Usually only either of uplink and downlink paths is active at one moment. However, both of the uplink and downlink paths will be active simultaneously when Baseband Front End is in loopback mode.

When either of TX windows in TDMA timer is opened, the uplink path in Baseband Front End will be activated. Accordingly components on the uplink path such as GMSK Modulator or 8PSK Modulator will be powered on, and then TX mixed-signal module is also powered on. The subblock Baseband Serial Ports will sink TX data bits from DSP and then forward them to GMSK Modulator or 8PSK Modulator. The outputs from GMSK Modulator or 8PSK Modulator are sent to TX mixed-signal module in format of I/Q signals. Finally D/A conversions are performed in TX mixed-signal module and the output analog signal is output to RF module. Additionally, 8PSK Modulation intrinsically extends the bursts window and reports in 8MVD (8PSK Modulation Valid) in BFE_STA status register.

Similarly, while either of RX windows in TDMA timer is opened, the downlink path in Baseband Front End will be activated. Accordingly components on the downlink path such as RX mixed-signal module and RX digital

 © 2009 MediaTek Inc.

FIR filter are then powered on. First A/D conversions are performed in RX mixed-signal module, and then the results in format of I/Q signals are sourced to Low Pass Filtering with different bandwidth (Narrow one about Fc = 90 khz, Wide one about FC = 110khz), Interference Detection Circuit to determine which Filter to be used by judging receiving power on current burst, Additionally, “I/Q Compensation Circuit” is an option in data path for modifying Receiving I/Q pair gain mismatch.. Finally the results will be sourced to DSP through Baseband Serial Ports.

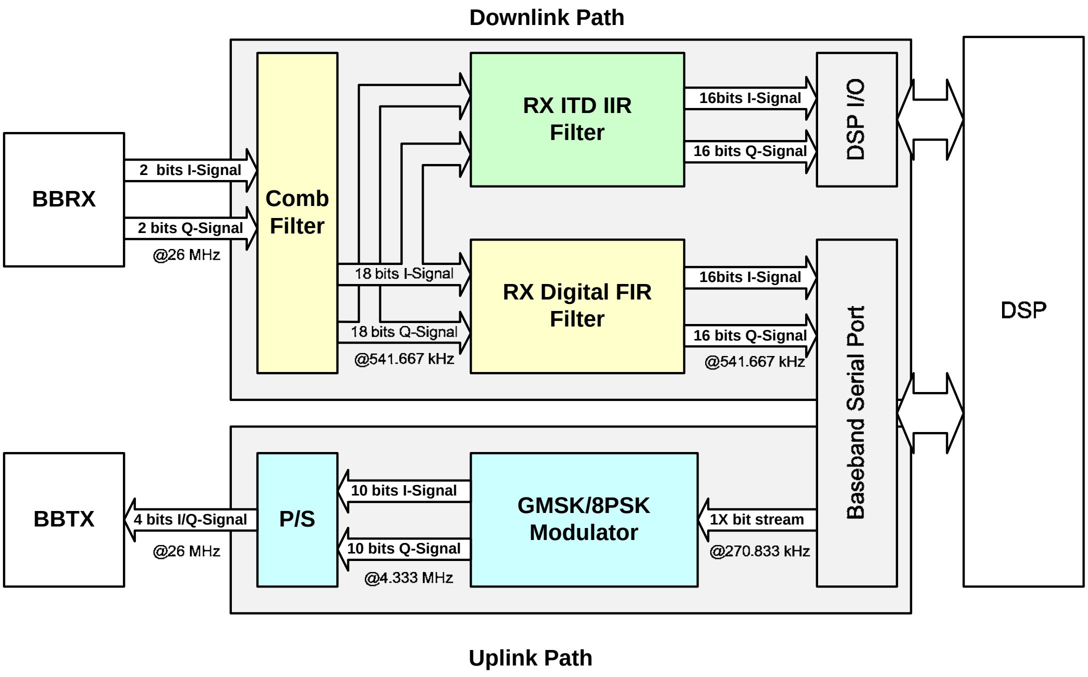


Figure 74 ***Block Diagram of Baseband Front End***

4.3.1 Baseband Serial Ports

4.3.1.1 General Description

Baseband Front End communicates with DSP through the sub block of Baseband Serial Ports. Baseband Serial Ports interfaces with DSP in serial manner. This implies that DSP must be configured carefully in order to have Baseband Serial Ports cooperate with DSP core correctly.

If downlink path is programmed in bypass-filter mode (**NOT** bypass-filter loopback mode), behavior of Baseband Serial Ports will be completely be different from that in normal function mode. The special mode is for testing purpose. Please see the subsequent section of Downlink Path for more details.

TX and RX windows are under control of TDMA timer. Please refer to functional specification of TDMA timer for the details on how to open/close a TX/RX window. Opening/Closing of TX/RX windows have two major effects on Baseband Front End: power on/off of corresponding components and data souring/sinking. It is worth noticing that Baseband Serial Ports is only intended for sinking TX data from DSP or sourcing data to DSP. It does not involve power on/off of TX/RX mixed-signal modules.

 © 2009 MediaTek Inc.

As far as downlink path is concerned, if a RX window is opened by TDMA timer Baseband Front End will have RX mixed-signal module proceed to make A/D conversion, two parallel RX digital filter proceed to perform filtering and Baseband Serial Ports be activated to source data from RX digital filter to Master DSP. However, the interval between the moment that RX mixed-signal module is powered on and the moment that data proceed to be dumped by Baseband Serial Ports can be well controlled in TDMA timer. Let us denote RX enable window as the interval that RX mixed-signal module is powered on and denote RX dump window as the interval that data is dumped by Baseband Serial Ports. If the first samples from RX digital filter desire to be discarded, the corresponding RX enable window must cover the corresponding RX dump window. Note that RX dump windows always win over RX enable windows. It means that a RX dump window will always raise a RX enable window. RX enable windows can be raised by TDMA timer or by programming RX power-down bit in global control registers to be ‘0’. This is useful in debugging environment.

Similarly, a TX dump window refers to the interval that Baseband Serial Ports sinks data from DSP on uplink path and a TX enable window refers to the interval that TX mixed-signal module is powered on. A TX window controlled by TDMA timer involves a TX dump window and a TX enable window simultaneously. The interval between the moment that TX mixed-signal module is powered on and the moment that data proceed to be forwarded from DSP to GMSK or 8PSK modulator by Baseband Serial Ports can be well controlled in TDMA timer. TX dump windows always win over TX enable windows. It means that a TX dump window will always raise a TX enable window. TX enable windows can be raised by TDMA timer or by programming TX power-down bit in global control registers to be ‘0’. It is useful in debugging environment.

Accordingly, Baseband Serial Ports are only under the control of TX/RX dump window. Note that if TX/RX dump window is not integer multiplies of bit-time it will be extended to be integer multiplies of bit-time. For example, if TX/RX dump window has interval of 156.25 bit-times then it will be extended to 157 bit-times in Baseband Serial Ports.

For uplink path, if uplink path is enabled, then the bit BULEN (Baseband Up-Link Enable) will be ‘1’. Otherwise the bit BULEN will be 0.

The MDSEL (Modulation Mode Select [3:0]) in TX_CONF control register needs to be latched in MDSEL shadow register according to the rising edge of TDMA Event Validate signal from TDMA controller, which used to indicate the modulation scheme selection between 8PSK or GMSK modulator for four transmit Burst.

Generally there will at most 4 sequential Bursts, 1st Burst, 2nd Burst, 3rd Bursts, and 4th Bursts, which are not necessary to be all turn on in a burst sequence. The BTXEN1, BTXEN2, BTXEN3, BTXEN4 will be asserted prior to each Bursts, and their rising edge will update the Mode selection control bit to select appropriate Modulation type for current input data symbols in each bursts. Additionally, this Mode selection status for each bursts will be stored in BFE_STA status register, including MDSTS1 (MoDulation mode StatuS1) , MDSTS2(MoDulation mode StatuS2), MDSTS3(MoDulation mode StatuS3), MDSTS4(MoDulation mode StatuS4), respectively.(Figure 75 Uplink Modulation Mode Selection Status Timing Diagram)

During these 4 bursts valid period, the bit BULFS (Baseband Uplink Frame Sync) in BFE_STA status register will be ‘1’. Otherwise will be ‘0’. Meanwhile, uplink path will forward TX bit from DSP to GMSK modulator or 8PSK

© 2009 MediaTek Inc.

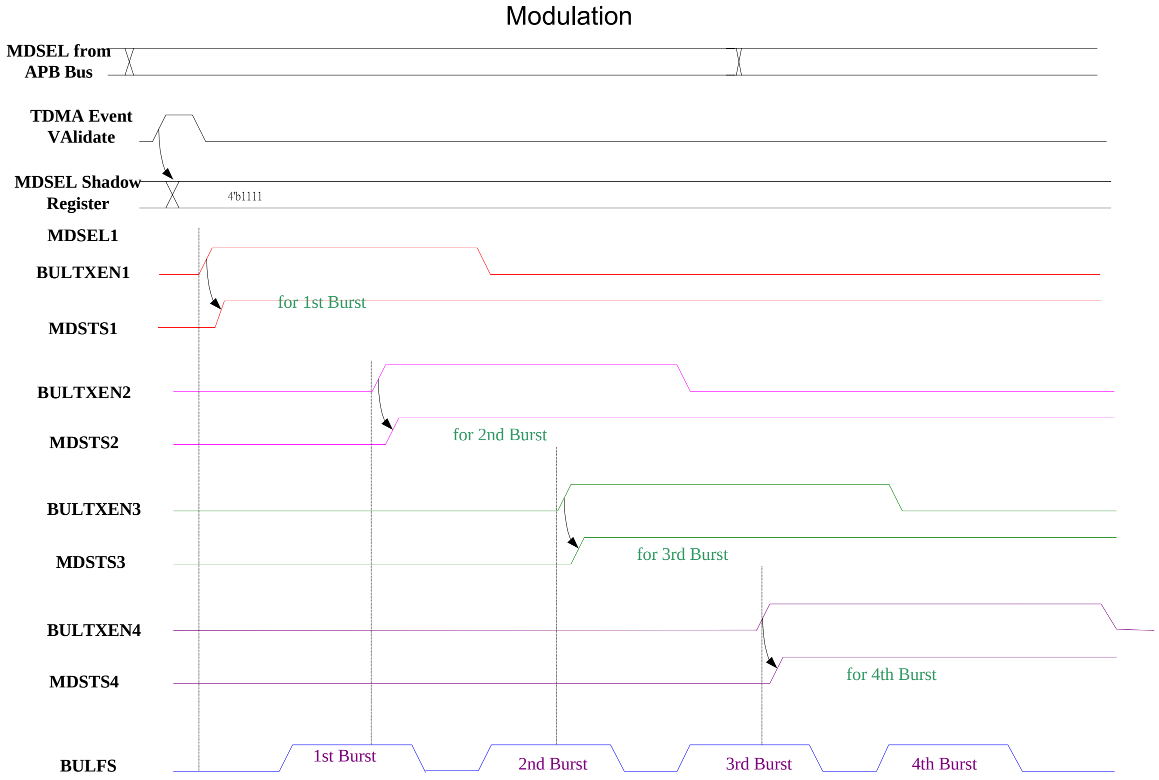


Figure 75 Uplink Modulation Mode Selection Status Timing Diagram

For downlink path, if BDLEN (Baseband DownLink Enable) is enabled, RX mixed-signal module will also be powered on. Similarly, once uplink path is enabled, TX mixed-signal module will also be powered on. Furthermore, enabling BDLFS (Baseband Down-Link FrameSync) Baseband Serial Ports for downlink path refers to dumping results from RX digital FIR filter to DSP.

4.3.1.2 Register Definitions

0x82100000 Base-band Common Control Register BFE_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**BCIEN**
Type																R/W
Reset																0

This register is for common control of Baseband Front End. It consists of ciphering encryption control.

BCIEN The bit is for ciphering encryption control. If the bit is set to ‘1’, XOR will be performed on some TX bits (payload of Normal Burst) and ciphering pattern bit from DSP, and then the result is forwarded to GMSK Modulator only. Meanwhile, Baseband Front End will generate signals to drive DSP ciphering process and produce corresponding ciphering pattern bits if the bit is set to ‘1’. If the bit is set to ‘0’, the TX bit from DSP will be forwarded to GMSK modulator directly. Baseband Front End will not activate DSP ciphering process.

0 Disable ciphering encryption.
1 Enable ciphering encryption.

0x82100004 Base-band Common Status Register BFE_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			MDSTS4	MDSTS3	MDSTS2	MDSTS1	BULEN4	BULEN3	BULEN2	BULEN1	BULFS4	BULFS3	BULFS2	BULFS1	BDLFS	BDLEN
Type			RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register indicates status of Baseband Front End This register indicates status of Baseband Front End. Under control of TDMA timer, Baseband Front End can be driven in several statuses. If downlink path is enabled, then the bit BDLEN will be '1'. Otherwise the bit BDLEN will be '0'. If downlink parts of Baseband Serial Ports is enabled, the bit BDLFS will be '1'. Otherwise the bit BDLFS will be '0'. If uplink path is enabled, then the bit BULEN will be '1'. Otherwise the bit BULEN will be 0. If uplink parts of Baseband Serial Ports is enabled, the bit BULFS will be '1'. Otherwise the bit BULFS will be '0'. Once downlink path is enabled, RX mixed-signal module will also be powered on. Similarly, once uplink path is enabled, TX mixed-signal module will also be powered on. Furthermore, enabling Baseband Serial Ports for downlink path refers to dumping results from RX digital FIR filter to DSP. Similarly, enabling Baseband Serial Ports for uplink path refers to forwarding TX bit from DSP to GMSK modulator. BDLEN stands for "**B**aseband **D**own**L**ink **En**able". BULEN stands for "**B**aseband **U**p**L**ink **En**able". BDLFS stands for "**B**aseband **D**own**L**ink **F**rame**S**ync". BULFS stands for "**B**aseband **U**p**L**ink **F**rame**S**ync".

BDLEN Indicate if downlink path is enabled.
0 Disabled
1 Enabled

BDLFS Indicate if Baseband Serial Ports for downlink path is enabled.
0 Disabled
1 Enabled

BULFS1 Indicate if Baseband Serial Ports for uplink path is enabled in 1st burst
0 Disabled
1 Enabled

BULFS2 Indicate if Baseband Serial Ports for uplink path is enabled in 2nd burst
0 Disabled
1 Enabled

BULFS3 Indicate if Baseband Serial Ports for uplink path is enabled in 3rd burst
0 Disabled
1 Enabled

BULFS4 Indicate if Baseband Serial Ports for uplink path is enabled in 4th burst
0 Disabled
1 Enabled

BULEN1 Indicate if uplink path is enabled in 1st burst.
0 Disabled
1 Enabled

BULEN2	Indicate if uplink path is enabled in 2nd burst.
0	Disabled
1	Enabled
BULEN3	Indicate if uplink path is enabled in 3rd burst.
0	Disabled
1	Enabled
BULEN4	Indicate if uplink path is enabled in 4th burst.
0	Disabled
1	Enabled
MDSTS1	Indicate the current Modulation Mode Selection in 1st burst
0	GMSK Modulation
1	8PSK Modulation
MDSTS2	Indicate the current Modulation Mode Selection in 2nd burst
0	GMSK Modulation
1	8PSK Modulation
MDSTS3	Indicate the current Modulation Mode Selection in 3rd burst
0	GMSK Modulation
1	8PSK Modulation
MDSTS4	Indicate the current Modulation Mode Selection in 4th burst
0	GMSK Modulation
1	8PSK Modulation

© 2009 MediaTek Inc.

4.3.2 Downlink Path (RX Path)

4.3.2.1 General Description

On the downlink path, the sub-block between RX mixed-signal module and Baseband Serial Ports is RX Path. It mainly consists of two parallel digital FIR filter with programmable tap number, two sets of multiplexing paths for loopback modes, interface for RX mixed-signal module, Interference Detection Circuit, I/Q Gain Mismatch compensation circuit, and interface for Baseband Serial Ports. The block diagram is shown in **Figure 76** Block Diagram of RX Path.

While RX enable windows are open, RX Path will issue control signals to have RX mixed-signal module proceed to make A/D conversion. As each conversion is finished, one set of I/Q signals will be latched. There exists a digital FIR filter for these I/Q signals. The result of filtering will be dumped to Baseband Serial Ports whenever RX dump windows are opened.

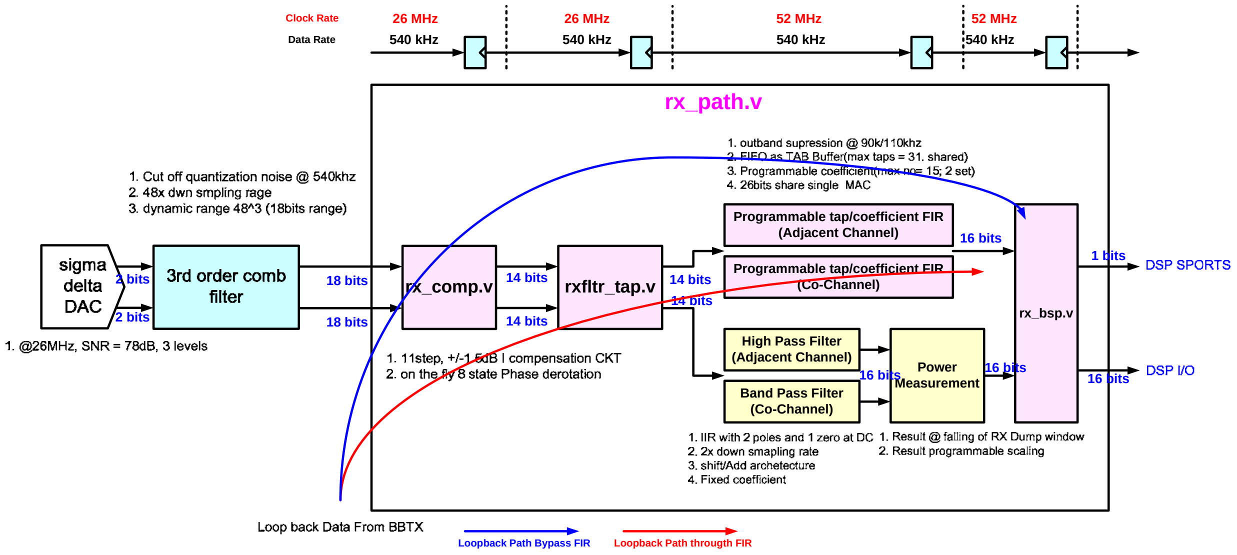


Figure 76 Block Diagram of RX Path

4.3.2.2 Comb Filter

The comb filter which takes the 2-bit A/D converter as input, and output the 18-bit I/Q data words to the baseband receiving path. The system is designed as 48X over-sampling with symbol period 541.7 kHz, thus the data inputs are 26MHz 2-bit signal. The input 2-bit signals are formed in (sign, magnitude) manner; that is, total 3 values are permitted as input: (-1, 0, +1).

The data path is mainly a decimation filter which contains the integration stages and the decimation stages. For a 3rd order design with 48X over-sampling, gain of the data path is 48^3 = 110592, which locates between 2^{16} and 2^{17}. Thus the internal word-length must be set to 18-bit to avoid overflow in the integration process.

 © 2009 MediaTek Inc.

4.3.2.3 Compensation Circuit - I/Q Gain Mismatch

In order to compensate I/Q Gain Mismatch , configure IGAINSEL(I Gain Selection) in RX_CON control register, the I over Q ratio can be compensate for 0.3 dB/step, totally 11 steps resulted in dynamic range up to +/-1.5dB.

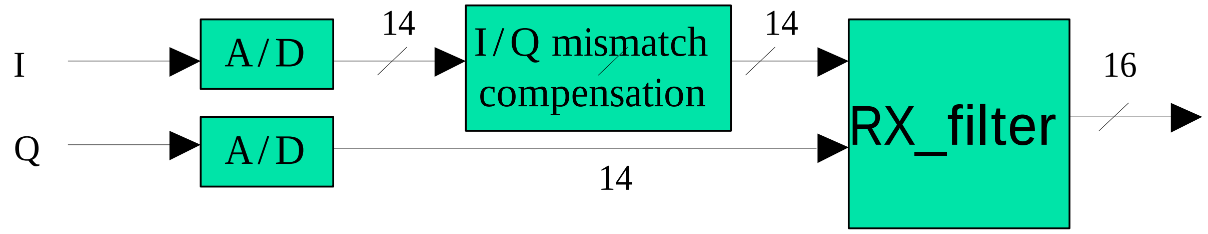


Figure 77 I/Q Mismatch Compensation Block Diagram

The I/Q swap functionality can be setting “1” for SWAP(I/Q Swapping) in RX_CFG control register, which is used to swap I/Q channel signals from RX mixed-signal module before they are latched into RX digital FIR filter. It is intended to provide flexibility for I/Q connection with RF modules

4.3.2.4 Phase De-rotation Circuit

Phase De-rotation Mode will usually turn on during FCB Detection for down conversion the wide spread receiving power to 67.7 kHz single tone.

Two separate control for implement this mode on data path through NarrowFIR filter or WideFIR filter by setting ‘1’ to PHROEN_N (Phase Rotate Enable for NarrowFIR) or PHROEN_W(Phase Rotate Enable for WideFIR) in RX_CON control register, respectively.

4.3.2.5 Adaptive Bandwidth & Programmable Digital FIR Filter

For the two parallel digital FIR Filter, the total tap number is programmable by FIRTPNO(FIR Tap number) in RX_CFG control register, which will configure the filter with different tap buffer depth.

4.3.2.5.1 Programmable tap & programmable Coefficient for FIR

In order to satisfy the signal requirements in both of idle and traffic modes, two sets of coefficients must be provided for the RX digital FIR filter. Therefore, the RX digital FIR filter is implemented as a FIR filter with programmable coefficients which can be accessed on the APB bus. The coefficient number can be programmable, range from 1~31. Each coefficient is ten-bit wide and coded in 2’s complement.

Take 21 Tap Coefficient for example, based on assumption that the FIR filter has symmetric coefficients, only 11 coefficients are implemented as programmable registers to save gate count. Denoting these digital filter coefficients as RX_RAM0_CS0 ~ RX_RAM0_CS11 (RX_RAM0 Coefficient Set 0~11), and these tap registers for I/Q channel signals as I/QTAPR [0:20], then the RX digital FIR filtering can be represented as the following equation:

$$I_{out}(n) = \sum_{i=0}^{20} \mathrm{BDLDFCR}[i] * \mathrm{ITAPR}[i] \Big|_{\text{at time } n+2m} = \mathrm{BDLDFCR}[11] * \mathrm{ITAPR}[11] + \sum_{i=0}^{11} \mathrm{BDLDFCR}[i] * \left(\mathrm{ITAPR}[i] + \mathrm{ITAPR}[20-i]\right)$$

$$Q_{out}(n) = \sum_{i=0}^{20} \mathrm{BDLDFCR}[i] * \mathrm{QTAPR}[i] \Big|_{\text{at time } n+2m} = \mathrm{BDLDFCR}[11] * \mathrm{QTAPR}[11] + \sum_{i=0}^{11} \mathrm{BDLDFCR}[i] * \left(\mathrm{QTAPR}[i] + \mathrm{QTAPR}[20-i]\right)$$

$$\mathrm{BDLDFCR}[i] = \mathrm{BDLDFCR}[20-i], i = 0, 1, \ldots, 11$$

where ITAPR [0] and QTAPR [0] are the latest samples for I- and Q-channel respectively and assume $I_{out}(0), Q_{out}(0)$ are obtained based on the content of tap registers at time moment *n*.

Additionally, the data sequence of two parallel FIR filter output will dump to Master DSP RX buffer in following order : "I channel output from Narrow FIR"=> " I channel output from Wide FIR""=>"Q channel output from Narrow FIR=>" Q channel output from Wide FIR.

4.3.2.5.2 Coefficient Set Selection

The Coefficient Set used for digital FIR can be changed during different burst mode switching. For example, during Normal Burst while no FB_STROBE (Frequency Burst Strobe, comes from TDMA controller) assertion, defined as "State B", "Coefficient Set ID" (CS ID) selection for both Narrow/Wide filter can be configured by ST_B_WCOF_SEL(State B Wide FIR Coefficient Selection) and "ST_B_NCOF_SEL" (State B Narrow FIR Coefficient Selection) on RX_FIR_CSID_CON control register, respectively. Usually during State B, Layer 1 software will select RAM table confidence from either RAM0 or RAM1 table in condition I for Narrow FIR and Wide FIR, respectively. The CS ID for both Narrow / Wide FIR filter be stored at Slave DSP RX buffer once TDMA trigger RX interrupt to DSP..ST_A_NCOF_SEL" (State A Narrow FIR Coefficient Selection) on RX_FIR_CSID_CON control register.
During FCB detection, MCU will notice TDMA controller by assertion FB_STROBE, defined as "StateA". "Coefficient Set ID" (CS ID) selection for both Narrow/Wide filter can be configured by ST_A_WCOF_SEL(State A Wide FIR Coefficient Selection) and "ST_A_NCOF_SEL" (State A Narrow FIR Coefficient Selection) on RX_FIR_CSID_CON control register, respectively. Usually during State B, Layer 1 software will select CS ID 2 and CSID 3 from either ROM0 or ROM1 table or RAM0 or RAM1 table in Condition II for Narrow FIR and Wide FIR, respectively.

4.3.2.5.3 Interference Detection Circuit for Adaptive Bandwidth Scheme

Used to compare the power of Co-channel Interference and Adjacent-channel Interference for determine if WideFIR filter is needed rather than default NarrowFIR filter. Two parallel path of power measurement for evaluating Co-channel effect or Adjacent Channel Effect by analyzing power after High Pass Filter (HPF) or Band Pass Filter (BPF), respectively. If Co-channel effect is worse than Adjacent Channel effect, WideFIR filter is needed.
The power measurement is accumulate I/Q Root Mean Square (RMS) power over the whole RX burst window, while exact accumulation period within the burst can be adjusted the starting point offset and duration length.. The "starting point Offset" and be configured by "RXID_PWR_OFF[7:0]" (RX Interference Detection Power Starting Point Offset) and duration period by " RXID_PWR_PER[7:0]"(RX Interference Detection Power Duration Period) in RX_PM_CON control register, while default value for starting offset is 11 and duration period is 141. The two accumulated power measurement output for Co-channel and Adjacent-channel will be received by DSP through DSP I/O port at the end of the duration period within a burst.
The power result can be further scale down by control the PWR_SHFT_NO(power right Shift Number) in RX_CON control register. E.g. set to "1" will divide the power output by two.

4.3.2.6 Debug Mode

4.3.2.6.1 Normal Mode bypass Filter

By setting "1" for BYPFLTR(Bypass Filter) in RX_CFG control register, the ADC outputs out of RX mixed-signal module will be directed into Baseband Serial Ports directly without through FIR. Limited by bandwidth of the serial interface between Baseband Serial Ports and DSP, only ADC outputs which are from either I-channel or Q-channel ADC can be dumped into DSP. Both I- and Q-channel ADC outputs cannot be dumped

simultaneously. Which channel will be dumped is controlled by the register bit SWAP of the control register RX_CFG when downlink path is programmed in “Bypass RX digital FIR filter” mode. See register definition below for more details. The mode is for measurement of performance of A/D converters in RX mixed-signal module.

4.3.2.6.2 TX-RX Digital Loopback Mode (Debug Mode)

In addition to normal function, there are two loopback modes in RX Path. One is bypass-filter loopback mode, and the other is through-filter loopback mode. They are intended for verification of DSP firmware and hardware. The bypass-filter loopback mode refers to that RX digital FIR filter is not on the loopback path. However, the through-filter loopback mode refers to that RX digital FIR filter is on the loopback path, while “ thru-Filter Loopback Mode” can be configured by setting “2’b10” for BLPEN(Baseband Loopback Enable) or “bypass-Filter Loopback Mode” by setting “ 2’b01” for BLPEN in RX_CON control register.

 © 2009 MediaTek Inc.

4.3.2.7 Register Definitions

4.3.2.7.1 APB Register

0x82100010 RX Configuration Register RX_CFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FIRTPNO								BYPFLTR	SWAP
Type							R/W								R/W	R/W
Reset							000000								0	0

This register is for configuration of downlink path, inclusive of configuration of RX mixed-signal module and RX path in Baseband Front End.

SWAP This register bit is for control of whether I/Q channel signals need to swap before they are inputted to Baseband Front End. It provides flexibility flexible of connection of I/Q channel signals between RF module and baseband module. The register bit has another purpose when the register bit "BYPFLTR" is set to 1. Please see description for the register bit "BYPFLTR".

0 I- and Q-channel signals are not swapped

1 I- and Q-channel signals are swapped

BYPFLTR Bypass RX FIR filter control. The register bit is used to configure Baseband Front End in the state called "Bypass RX FIR filter state" or not. Once the bit is set to '1', RX FIR filter will be bypassed. That is, ADC outputs of RX mixed-signal module that are has 14-bit resolution and at sampling rate of 571 kHz can be dumped into DSP by Baseband Serial Ports and RX FIR filtering will not be performed on them.

0 Not bypass RX FIR filter

1 Bypass RX FIR filter

FIRTPNO RX FIR filter tap no. select. This control register will control the two parallel digital filter with different tap buffer depth since the FIR function in symmetric behavior. The maximum tap number is 31, minimum is 1. ODD number only.

00001 1 tap

00011 3 tap

- •
- •
- •

11101 29 tap

11111 31 tap

0x82100014 RX Control Register RX_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					PWR_SHFT_NO				IGAINSEL				PH_ROEN_N	PH_ROEN_W	BLPEN	
Type					R/W				R/W				R/W	R/W	R/W	
Reset					0000				0000				0	0	00	

This register is for control of downlink path, inclusive of control of RX mixed-signal module and RX path in Baseband Front End module.

BLPEN The register field is for loopback configuration selection in Baseband Front End.

- **00** Configure Baseband Front End in normal function mode
- **01** Configure Baseband Front End in bypass-filter loopback mode
- 10 Configure Baseband Front End in through-filter loopback mode
- 11 Reserved

PH_ROEN_W Enable for I/Q pair Phase De-rotation in Wide FIR Data Path,

- **0** Disable Phase De-rotation for I/Q pair
- **1** Enable Phase De-rotation for I/Q pair

PH_ROEN_N Enable for I/Q pair Phase De-rotation in Narrow FIR Data Path,

- **0** Disable Phase De-rotation for I/Q pair.
- **1** Enable Phase De-rotation for I/Q pair

IGAINSEL RX I data Gain Compensation Select. 0.3dB/step, totally 11 steps and dynamic range up to +/-1.5dB for

- **0000** compensate 0dB for I/Q
- **0001** compensate 0.3dB for I/Q
- **0010** compensate 0.6dB for I/Q
- **0011** compensate 0.9dB for I/Q
- **0100** compensate 1.2dB for I/Q
- **0101** compensate 1.5dB for I/Q
- **1001** compensate –0.3dB for I/Q
- **1010** compensate -0.6dB for I/Q
- **1011** compensate –0.9dB for I/Q
- **1100** compensate –1.2dB for I/Q
- **1101** compensate –1.5dB for I/Q
- **Default** no compensation for I/Q

PWR_SHFT_NO Power measuring Result Right Shift Number. The Power level measurement result can be right shift from 0 to 15 bits.

0x82100018 RX Interference Detection Power Measurement Control Register RX_PM_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RXID_PWR_PER								RXID_PWR_OFF							
Type	R/W								R/W							
Reset	0x8D								0xb							

RXID_PWR_OFF RX Interference Detection Power Measurement Starting Offset. Setting this register will delay the starting time of Interference Detection Power Measurement in symbol time unit. Maximum value is 156, while default value is 11 (0xB).

RXID_PWR_PER RX Interference Detection Power Measurement Accumulation Period. By setting this control register will determine the length of accumulation duration for power Measurement. Minimum value is 0, Maximum value is 156, while default value is 141(0x8D). Please notice that RXID_PWR_OFF + RXID_PWR_PER should less than 154 due to hardware implementation limitation.

0x8210001C RX FIR Coefficient Set ID Control Register RX_FIR_CSID_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ST_A_NCOF_SEL								ST_B_NCOF_SEL				ST_B_WCOF_SEL			
Type	R/W								R/W				R/W			
Reset	0000								0010				0011			

These three set of Coefficient Set ID will be dump to slave DSP RX Buffer for indicating the current selection of FIR coefficient from either RAM or ROM table, while CSID= 0 represents ROM table selection, and CSID2~CSID15 represent RAM table selection.

ST_B_WCOF_SEL State B Coefficient Set Selection for Wide FIR.

ST_B_NCOF_SEL State B Coefficient Set Selection for Narrow FIR.

ST_A_NCOF_SEL State A Coefficient Set Selection for Narrow FIR.

0x82100070 RX RAM0Coefficient Set 0Register RX_RAM0_CS0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							RX_RAM0_CS0									
Type							R/W									
Reset							000000000									

This register is 1st of the 16 coefficient in RAM0 table, Coefficient Set ID 2 or 4. The content is coded in 2's complement. That is, its maximum is 255 and its minimum is –256, while the total coefficient number in this Coefficient Set has to be greater than half of TAPNO(programmable Tap no.) and smaller than half of maximum tap no(15).

Register Address	Register Function	Acronym
0x82100070	RX RAM0Coefficient Set 0 Register	RX_RAM0_CS0
0x82100074	RX RAM0Coefficient Set 1 Register	RX_RAM0_CS1
0x82100078	RX RAM0Coefficient Set 2 Register	RX_RAM0_CS2
0x8210007C	RX RAM0Coefficient Set 3 Register	RX_RAM0_CS3
0x82100080	RX RAM0Coefficient Set 4 Register	RX_RAM0_CS4
0x82100084	RX RAM0Coefficient Set 5 Register	RX_RAM0_CS5
0x82100088	RX RAM0Coefficient Set 6 Register	RX_RAM0_CS6
0x8210008C	RX RAM0Coefficient Set 7 Register	RX_RAM0_CS7
0x82100090	RX RAM0Coefficient Set 8 Register	RX_RAM0_CS8
0x82100094	RX RAM0Coefficient Set 9 Register	RX_RAM0_CS9
0x82100098	RX RAM0Coefficient Set 10 Register	RX_RAM0_CS10
0x8210009C	RX RAM0Coefficient Set 11Register	RX_RAM0_CS11
0x821000a0	RX RAM0Coefficient Set 12Register	RX_RAM0_CS12
0x821000a4	RX RAM0Coefficient Set 13Register	RX_RAM0_CS13
0x821000a8	RX RAM0Coefficient Set 14 Register	RX_RAM0_CS14
0x821000aC	RX RAM0Coefficient Set 15 Register	RX_RAM0_CS15

0x82100020 RX RAM1 Coefficient Set 0 Register RX_RAM1_CS0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**RX_RAM1_CS0**									
Type							R/W									
Reset							000000000									

This register is 1st of the 16 coefficient in RAM1 table, Coefficient Set ID 2 or 4. The content is coded in 2's complement. That is, its maximum is 255 and its minimum is –256, while the total coefficient number in this Coefficient Set has to be greater than half of TAPNO(programmable Tap no.) and smaller than half of maximum tap no(15).

Register Address	Register Function	Acronym
0x82100020	RX RAM1 Coefficient Set 0 Register	RX_RAM1_CS0
0x82100024	RX RAM1 Coefficient Set 1Register	RX_RAM1_CS1
0x82100028	RX RAM1 Coefficient Set 2 Register	RX_RAM1_CS2
0x8210002C	RX RAM1 Coefficient Set 3 Register	RX_RAM1_CS3
0x82100030	RX RAM1 Coefficient Set 4 Register	RX_RAM1_CS4
0x82100034	RX RAM1 Coefficient Set 5 Register	RX_RAM1_CS5
0x82100038	RX RAM1 Coefficient Set 6 Register	RX_RAM1_CS6
0x8210003C	RX RAM1 Coefficient Set 7 Register	RX_RAM1_CS7
0x82100040	RX RAM1 Coefficient Set 8 Register	RX_RAM1_CS8
0x82100044	RX RAM1 Coefficient Set 9 Register	RX_RAM1_CS9
0x82100048	RX RAM1 Coefficient Set 10 Register	RX_RAM1_CS10
0x8210004C	RX RAM1 Coefficient Set 11 Register	RX_RAM1_CS11
0x82100050	RX RAM1 Coefficient Set 12 Register	RX_RAM1_CS12
0x82100054	RX RAM1 Coefficient Set 13 Register	RX_RAM1_CS13
0x82100058	RX RAM1 Coefficient Set 14 Register	RX_RAM1_CS14
0x8210005C	RX RAM1 Coefficient Set 15 Register	RX_RAM1_CS15

0x821000B0 RX Interference Detection HPF Power Register RX_HPWR_STS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RX_PWR_HPF**															
Type	R/O															
Reset	0000000000000000															

This register is for read the power measurement result of the HPF interference detection filter.

RX_PWR_HPF Value of the power measurement result for the outband interference detection.

0x821000B4 RX Interference Detection BPF Power Register RX_BPWR_STS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RX_PWR_BPF**															
Type	R/O															
Reset	0000000000000000															

 © 2009 MediaTek Inc.

This register is for read the power measurement result of the BPF interference detection filter.

RX_PWR_BPF Value of the power measurement result for the inband interference detection

4.3.2.7.2 DSP I/O Register

0x743 RX HPF ITD Power Register of Window0 DSPIO_ITD_H_0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ITD_H_DATA_0															
Type	R/O															
Reset	0000000000000000															

This register is for **DSP** to read the power measurement result of the BPF interference detection filter through DSP I/O.

DSPIO_ITD_H_0 Value of the power measurement result for the outband interference detection of window 0.

Register Address	Register Function	Acronym
0x743	RX HPF ITD Power Register of Window0	DSPIO_ITD_H_0
0x747	RX HPF ITD Power Register of Window1	DSPIO_ITD_H_1
0x74B	RX HPF ITD Power Register of Window2	DSPIO_ITD_H_2
0x74F	RX HPF ITD Power Register of Window3	DSPIO_ITD_H_3
0x753	RX HPF ITD Power Register of Window4	DSPIO_ITD_H_4
0x757	RX HPF ITD Power Register of Window5	DSPIO_ITD_H_5

0x744 RX BPF ITD Power Register of Window0 DSPIO_ITD_B_0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ITD_B_DATA_0															
Type	R/O															
Reset	0000000000000000															

This register is for **DSP** to read the power measurement result of the BPF interference detection filter through DSP I/O.

DSPIO_ITD_B_0 Value of the power measurement result for the inband interference detection of window 0.

Register Address	Register Function	Acronym
0x744	RX BPF ITD Power Register of Window0	DSPIO_ITD_B_0
0x748	RX BPF ITD Power Register of Window1	DSPIO_ITD_B_1
0x74C	RX BPF ITD Power Register of Window2	DSPIO_ITD_B_2
0x750	RX BPF ITD Power Register of Window3	DSPIO_ITD_B_3
0x754	RX BPF ITD Power Register of	DSPIO_ITD_B_4

	Window4	
0x758	RX BPF ITD Power Register of Window5	DSPIO_ITD_B_5

0x759 RX ITD Power Measurement Ready Flag DSPIO_RXID_RDY

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											RXID_RDY_5	RXID_RDY_4	RXID_RDY_3	RXID_RDY_2	RXID_RDY_1	RXID_RDY_0
Type											R/O	R/O	R/O	R/O	R/O	R/O
Reset											0	0	0	0	0	0

This register is for **DSP** to see whether the RX ITD power register is ready or not through DSP I/O. When the DSPIO_ITD_H_0 and DSPIO_ITD_B_0 are ready, bit 0 is set to 1. Moreover, while DSP read the data of DSPIO_ITD_H_0 and DSPIO_ITD_B_0, bit 0 is reset to 0.

RXID_RDY_0 Ready flag for DSP to read the ITD power measurement result of window0.
RXID_RDY_1 Ready flag for DSP to read the ITD power measurement result of window1.
RXID_RDY_2 Ready flag for DSP to read the ITD power measurement result of window2.
RXID_RDY_3 Ready flag for DSP to read the ITD power measurement result of window3.
RXID_RDY_4 Ready flag for DSP to read the ITD power measurement result of window4.
RXID_RDY_5 Ready flag for DSP to read the ITD power measurement result of window5.

4.3.3 Uplink Path (TX Path)

4.3.3.1 General Description

The purpose of the uplink path inside Baseband Front End is to sink TX symbols, from DSP, then perform GMSK modulation or 8PSK Modulation on them, then perform offset cancellation on I/Q digital signals, and finally control TX mixed-signal module to make D/A conversion on I/Q signals out of GMSK Modulator or 8PSK Modulator with offset cancellation. Accordingly, the uplink path is composed of uplink parts of Baseband Serial Ports, GSM Encryptor, GMSK Modulator, 8PSK Modulator and several compensation circuit including I/Q DC offset, I/Q Quadrature Phase Compensation, and I/Q Gain Mismatch. The block diagram of uplink path is shownas followed.

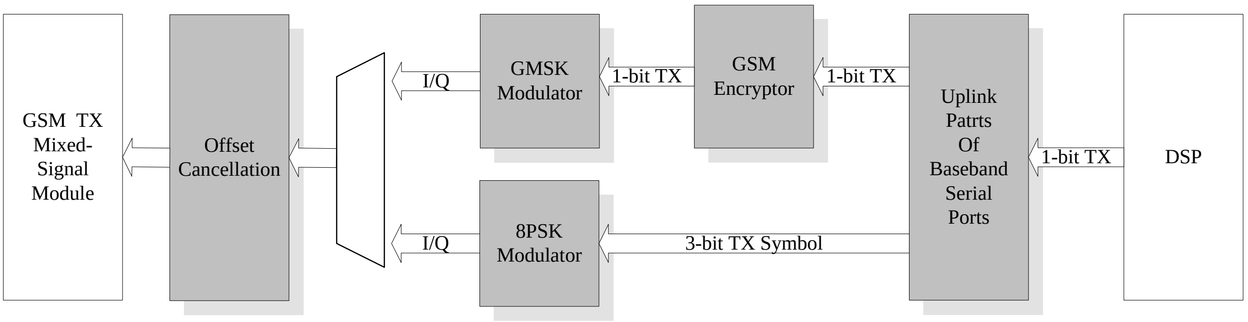


Figure 78 Block Diagram of Uplink Path

On uplink path, the content of a burst, including tail bits, data bits, and training sequence bits is sent from DSP. DSP outputs will be t Translated by either GMSK Modulator or 8PSK Modulator. The Modulation Mode

Selection is controlled by MDSEL1 (Modulation Mode Select1) MDSEL2, MDSEL3, MDSEL4 in TX_CFG control register, and these translated bits after modulation will become I/Q digital signals with certain latency.
TDMA timer having a quarter-bit timing accuracy gives the timing windows for uplink operation. Uplink operation is controlled by TX enable window and TX dump window of TDMA timer. Usually, TX enable window is opened earlier than TX dump window. When TX enable window of TDMA timer is opened, uplink path in Baseband Front End will power-on GSK TX mixed-signal module and thus drive valid outputs to RF module. However, uplink parts of Baseband Serial Ports still do not sink data from DSP through the serial interface between Baseband Serial Ports and DSP until TX dump window of TDMA timer is opened.

4.3.3.2 Compensation Circuit

4.3.3.2.1 Quadrature Phase

For 8PSK Modulation, in order to improve the EVM performance, use PHSEL[3:0](Phase Select) in TX_CFG control register to compensate the quadrature phase. 10 steps, 1degree/step, up to +/5 degree dynamic range.

4.3.3.2.2 DC offset Cancellation

Offset cancellation will be performed on these I/Q digital signals to compensate offset error of D/A converters (DAC) in TX mixed-signal module. Finally the generated I/Q digital signals will be input to TX mixed-signal module that contains two DAC for I/Q signal respectively.

4.3.3.3 Auxiliary Calibration Circuit - 540khz Sine Tone Generator

By setting ‘1’ to SGEN(Sine Tone Generation) in TX_CFG control register, the BBTX output will become 540khz single sine tone, which is used for Factory Calibration scheme for Mixed Signal Low Pass Filter Cut-off Frequency Accuracy.

4.3.3.4 GSM Encryptor

When uplink parts of Baseband Serial Ports pass a TX symbol to GSM Encryptor, GSM Encryptor will perform encryption on the TX symbol if set ‘1’ to BCIEN(Baseband Ciphering Encryption) in **BFE_CON** register. Otherwise, the TX symbol will be directed to GMSK modulator directly.

4.3.3.5 Modulation

4.3.3.5.1 GMSK Modulation

GMSK Modulator is used to convert bit stream of GSM bursts into in-phase and quadrature-phase outputs by means of GMSK modulation scheme. It consists of a ROM table, timing control logic and some state registers for GMSK modulation scheme. GMSK Modulator is activated when TX dump window is opened. There is latency between assertion of TX dump window and the first valid output of GMSK Modulator. The reason is because the bit rate of TX symbols is 270.833 KHz and the output rate of GMSK Modulator is 4.333 MHz, and therefore timing synchronization is necessary between the two rates.
Additionally, in order to prevent phase discontinuity in between the multiple-burst Mode, the GMSK modulator will output continuous 67.7khs sine tone outside the burst once RX DAC Enable window is still asserted. Once RX DAC Enable window is disserted, GMSK modulator will park at DC level.

4.3.3.5.2 8PSK Modulation

8PSK Modulator is used to convert bit stream of EDGE bursts into basically 8 phase I/Q pair output by means of 8PSK modulation scheme. It consists of ROM table, timing control logic and some state registers for 8PSK

modulation scheme. The conversion is based on 5 sequential symbol and performed moving average from the ROM table lookup. 8PSK Modulator is activated when TX dump window is opened. There is one clock delay between assertion of TX dump window and the first valid output of 8PSK Modulator. The reason is because the bit rate of TX symbols is 270.833 KHz and the output rate of 8PSK Modulator is 4.333 MHz, and therefore timing synchronization is necessary between the two rates.

4.3.3.5.2.1 8PSK Ramp Profile

During 8PSKModulation, there will be 3 Ramp Profile to select to choose the BBTX I/Q output during the guard period, where the DAC_ON is asserted while TX_WINDOW is de-asserted. This control register is an option to adjust the transmitter performance on “Modulation Transient Spectrum” requirement of ETSI SPEC if different companion Power Amplifier solution is chosen

By setting RPSEL (Ramp Profile Select) in TX_CFG control register to ‘0’will configured 8PSK Modulator to Ramp Profile I and I/Q output will be about 50 kHz Sine-tone before the first rising edge of BULFS, after the last falling edge of BULFS, and in between the bursts. For Ramp Profile II, BBTX I/Q output will be quiescent low DC (null-DC) level during the guard period.

For Ramp Profile III, initial guard period will be 50 kHz sine-tone, while the reset guard period will be null-DC level.

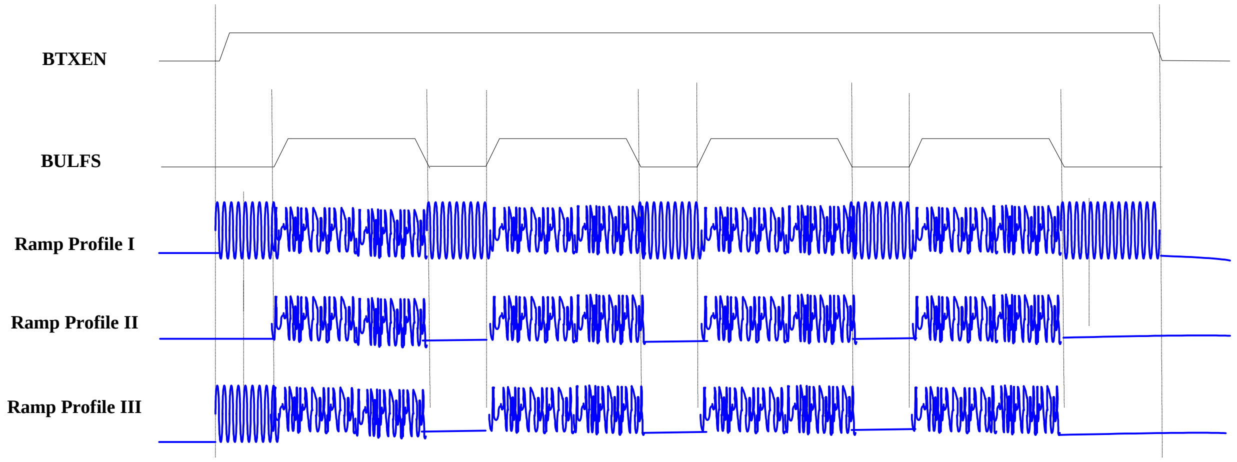


Figure 79 Ramp Profile I/II/III in 8PSK Modulation for Multi-Bursts configuration..

4.3.3.5.3 I/Q Swap

By setting ‘1’ to IQSWP in TX_CFG control register, phase on I/Q plane will rotate in inverse direction. This option is to meet the different requirement form RF chip regarding I/Q plane. This control signal is for GMSK Modulation only.

4.3.3.5.4 Modulation Output Latency Adjustment

For Multiple bursts, there maybe are consecutive bursts with different modulation mode. (E.g. switch GMSK to 8PSK or vice versa). However, there are about 8 to 10 QB output latencies for either GMSK/8PSKmodulation output. In order to match the transition timing of power ramp control in the power amplifier outside the baseband chip, we have to precisely control the SW_QBCNT (modulation Switching Quarter Bit CouNT) in

TX_CFG control register. , which will program the mode switching timing in QB count unit during the inter-slot period. Normally the inter-slot period is about 33 QB Count, and the default value to switch the modulation mode is 24 QB count (8 QB count after the middle point)

Additionally, by programming GMSK_DTAP_SYM(GMSK Delay Tap) in TX_CFG and GMSK_DTAP_QB in TX_CON control register, the output latency for GMSK modulation output can be adjust to compensate the offset between GMSK/8PSK modulator. The GMSK_DTAP_SYM adjust the output latency in symbol time(3.69us), while GMSK_DTA_QB adjust in Quarter Bit(QB) Time (0.92us).Default value is delay 1 symbol (3.69us) of GMSK modulator output.

4.3.3.5.5 Modulation Mode Switching

By setting '1' to INTEN(Interpolation Enable) in TX_CFG control register, if two consecutive bursts belongs to 8PSK Modulation and GMSK Modulation, or vice versa, 32 steps interpolation between two Modulator outputs for 4quater bit long in guard period..

4.3.3.5.6 Debug Mode

4.3.3.5.6.1 Modulation Bypass Mode

For DSP debug purpose, set both '1' for MDBYP(Modulator Bypass) in TX_CFG control register and BYPFLR(Bypass RX Filter) in RX_CFG control register for directly loopback DSP 16-bits data (10bits valid data plus sign or zero extension) through DAC only.

4.3.3.5.6.2 Force GMSK/8PSK Modulator turn on

By setting '1' to APNDEN(Append Enable) bit in TX CFG control register, both GMSK and 8PSK modulator will park on constant DC level during the non-burst period, while the I/Q pair output phase maybe discontinuous since both modulator will be reset at the beginning of the burst. However, the reset of the modulator will be helpful for the debugging purpose.

4.3.3.6 Register Definitions

0x82100060 TX Configuration Register TX_CFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GMSK_DTAP_SYM		SW_QBCNT					ALL_10_EN		SGEN	MDBYP	INTEN	RPSEL		APNDEN
Type		R/W		RW					RW		R/W	R/W	R/W	R/WR/W		R/W
Reset		00		11000					00		0	0	0	00		0

This register is for configuration of uplink path, inclusive of configuration of TX mixed-signal module and TX path in Baseband Front End.

APNDEN Appending Bits Enable.(For DSP digital loopback debug mode) The register bit is used to control the ending scheme of GPRS Mode GMSK modulation only.

0 Suitable for GPRS /EDGE mode. If a TX enable window contains several TX dump window, then GMSK modulator will still output in the intervals between two TX dump window and all 1's will be fed into GMSK modulator. In the other word, mainly used PA to perform the power ramp up/down, while Modulator output low amplitude sinewave. **Note that when the bit is set to '0', the**

interval between the moment at which TX enable window is activated and the moment at which TX dump window is activated must be multiples of one bit time.

1 Suitable for GSM only. After a TX dump window, GMSK modulator will only output for some bit time.

RPSEL Ramp Profile Select for 8PSK Modulation. The register bit is used to select either Ramp Profile I / Ramp Profile II for EDGE Mode 8PSK Modulation only.

0 Ramp Profile I. Generate 50Khz sine tone during the guard period among BBTX bursts by repeated input pattern [7 7 7 7]

1 Ramp Profile II. Generate null DC I/Q output during guard period among BBTX bursts

2 Reserved

3 Ramp Profile III , Generate 50 kHz sine tone after DAC_ON asserted and before TX_WIDNOW asserted if 1st burst is 8PSK modulation, while the reset guard period always output null DC I/Q output. If the 1st burst is GMSK modulation, the I/Q output will be always null DC as Ramp Profile II.

INTEN Interpolation Enable. During Multi-bursts Mode, if two consecutive bursts belongs to 8PSK Modulation and GMSK Modulation, or vice versa, set this bit to select either takes 32 steps interpolation between two Modulator outputs in guard period..

0 Regular Transition Mode.

1 Interpolation Transition Mode.

MDBYP Modulator Bypass (For DSP Debug Mode) Select. The register bit is used to select the bypass mode for I/Q pair outputs bypassed both the GMSK/8PSK modulator

0 Regular Modulation Mode

1 Bypass Modulator Mode (DSP Debug Mode).

SGEN SineTone Generator Enable.(For Factory Calibration Purpose). The register bit is used to select the TX modulator output switch to 540 kHz Sine Tone.

0 BBTX output from regulator modulator output.

1 BBTX output switch to 540 kHz sine Tone.

ALL_10GEN For Debug mode of BBTX. Generate all 1's or zero's input during BBTX valid burst. For GMSK modulation, set 2'b1 or 2'b10 will generate 67.7 kHz sine tone, while 8PSK modulator will generate 50khz sine tone. Default value 2'b00 is normal mode.

0 Normal Mode, regular modulator input from Slave DSP TX Buffer.

1 Debug Mode, All zero's input pattern generated; GMSK modulator will generate 67.7 kHz sine tone. 8PSK modulator will generate 50 kHz sine tone.

2 Debug Mode All 1's input pattern generated; GMSK modulator will generate 67.7 kHz sine tone. 8PSK modulator will generate 50 kHz sine tone.

SW_QBCNT Control the mode switching timing in the inter-slot period in Quarter Bit Count for modulation mode switching in multiple bursts. Normally the inter-slot period is about 33 QB Count, and the default value to switch the modulation mode is 24 QB count (8 QB count after the middle point). Program range from "5~31", while default value is 24.

GMSK_DTAP_SYM Control the GMSK modulator output latency in symbol time (3.69us/symbol) in order to match the output latency offset between 8PSK /GMSK modulator

0 Delay 1 TAP for GMSK modulator output

1 No delay for GMSK modulator output

2 Delay 2 TAP for GMSK modulator output

0x82100064 TX Control Register TX_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			GMSK_DTAP_QB		PHSEL						MDSEL4	MDSEL3	MDSEL2	MDSEL1		IQSWP
Type			R/W		R/W						R/W	R/W	R/W	R/W		R/W
Reset			00		0000						0	0	0	0		0

This register is for control of uplink path, inclusive of control of TX mixed-signal module and TX path in Baseband Front End.

IQSWP The register bit is for control of I/Q swapping. When the bit is set to ‘1’, phase on I/Q plane will rotate in inverse direction. Moreover, this register is double buffered by EVENT_VALIDATE.

- **0** I and Q are not swapped.
- **1** I and Q are swapped.

MDSEL1 Modulation Mode Select for 1st Burst. The register bit is used to select either GMSK or 8PSK Modulation for GSM/GPRS mode or EDGE mode.

- **0** GMSK Modulation for GSM/GPRS mode.
- **1** 8PSK Modulation for EDGE mode.

MDSEL2 Modulation Mode Select for 2nd Burst. The register bit is used to select either GMSK or 8PSK Modulation for GSM/GPRS mode or EDGE mode.

- **0** GMSK Modulation for GSM/GPRS mode.
- **1** 8PSK Modulation for EDGE mode.

MDSEL3 Modulation Mode Select for 3rd Burst. The register bit is used to select either GMSK or 8PSK Modulation for GSM/GPRS mode or EDGE mode.

- **0** GMSK Modulation for GSM/GPRS mode.
- **1** 8PSK Modulation for EDGE mode.

MDSEL4 Modulation Mode Select for 4th Burst. The register bit is used to select either GMSK or 8PSK Modulation for GSM/GPRS mode or EDGE mode.

- **0** GMSK Modulation for GSM/GPRS mode.
- **1** 8PSK Modulation for EDGE mode.

PHSEL Quadrature phase compensation select

- **0000**: 0 degree compensation.
- **0001**: 1 degree compensation.
- **0010**: 2 degree compensation.
- **0011**: 3 degree compensation.
- **0100**: 4 degree compensation.
- **0101**: 5 degree compensation.
- **1010**: -5 degree compensation.
- **1011**: -4 degree compensation.
- **1100**: -3 degree compensation.
- **1101**: -2 degree compensation.
- **1110**: -1 degree compensation.
- **1111**: 0 degree compensation.

GMSK_DTAP_QB Control the GMSK modulator output latency in Quarter Bit(QB) Time (0.92us/QB) in order to match the output latency offset between 8PSK /GMSK modulator

- **0** No Delay GMSK modulator output
- **1** Delay 1QB for GMSK modulator output
- **2** Delay 2 QB for GMSK modulator output
- **3** Delay 3QB for GMSK modulator output

0x82100068 TX I/Q Channel Offset Compensation Register TX_OFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OFF_TYP		OFFQ[5:0]								OFFI[5:0]					
Type	R/W		R/W								R/W					
Reset	0		000000								000000					

The register is for offset cancellation of I-channel DAC in TX mixed-signal module. It is for compensation of offset error caused by I/Q-channel DAC in TX mixed-signal module. It is coded in 2's complement, that is, with maximum 31 and minimum –32.

OFFI Value of offset cancellation for I-channel DAC in TX mixed-signal module (+31mV → -32mV)

OFFQ Value of offset cancellation for Q-channel DAC in TX mixed-signal module (+31mV → -32mV)

OFF_TYP Type of the OFFI and OFFQ register. While OFF_TYP = 1, the offset values are double buffered and can be chaneged burst by burst after EVENT_VALIDATE comes. Otherwise, the offset values would change immidately after the coming of APB commands, which can't be adjusted burst by burst.

- **0** No double buffer
- **1** Double buffered

4.4 Baseband Parallel Interface

4.4.1 General Description

The Baseband Parallel Interface features 10 control pins, which are used for timing-critical external circuits. These pins typically control front-end components which must be turned on or off at specific times during GSM operation, such as transmit-enable, band switching, TR-switch, etc.

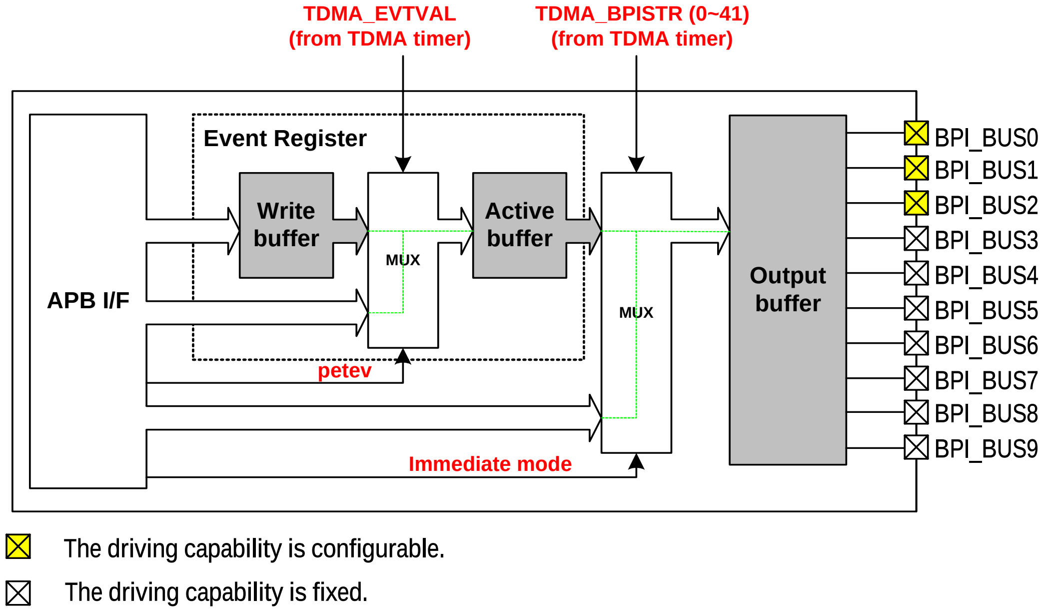


***Figure 80** Block diagram of BPI interface*

The user can program 42 sets of 10-bit registers to set the output value of BPI_BUS0~BPI_BUS9. The data is stored in the write buffers. The write buffers are then forwarded to the active buffers when the TDMA_EVTVAL signal is pulsed, usually once per frame. Each of the 42 write buffers corresponds to an active buffer, as well as to a TDMA event.

Each TDMA_BPISTR event triggers the transfer of data in the corresponding active buffer to the output buffer, thus changing the value of the BPI bus. The user can disable the events by programming the enable registers in the TDMA timer. If the TDMA_BPISTR event is disabled, the corresponding signal TDMA_BPISTR is not pulsed, and the value on the BPI bus remains unchanged.

For applications in which BPI signals serve as the switch, current-driving components are typically added to enhance driving capability. Three configurable output pins provide current up to 8 mA, and help reduce the number of external components. The output pins BPI_BUS6, BPI_BUS7, BPI_BUS8, and BPI_BUS9 are multiplexed with GPIO. Please refer to the GPIO table for more detailed information.

4.4.2 Register Definitions

Register Address	Register Function	Acronym
0x82020000	BPI control register	BPI_CON
0x820200B0	BPI event enable register 0	BPI_ENA0
0x820200B4	BPI event enable register 1	BPI_ENA1
0x820200B8	BPI event enable register 2	BPI_ENA2

Table 70 ***BPI Control Registers***

0x82020000 BPI control register BPI_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**PINM2**	**PINM1**	**PINM0**	**PETEV**
Type													WO	WO	WO	R/W
Reset													0	0	0	0

This register is the control register of the BPI unit. The register controls the direct access mode of the active buffer and the current driving capability for the output pins.

The driving capabilities of BPI_BUS0, BPI_BUS1 and BPI_BUS2 can be 2 mA or 8 mA, determined by the value of PINM0, PINM1 and PINM2 respectively. These output pins provide a higher driving capability and save on external current-driving components. In addition to the configurable pins, pins BPI_BUS3 to BPI_BUS9 provide a driving capability of 2 mA (fixed).

PETEV Enables direct access to the active buffer.

- **0** The user writes data to the write buffer. The data is latched in the active buffer after the TDMA_EVTVAL signal is pulsed.
- 1 The user directly writes data to the active buffer without waiting for the TDMA_EVTVAL signal.

PINM0 Controls the driving capability of BPI_BUS0.

- **0** The output driving capability is 2mA.
- **1** The output driving capability is 8mA.

PINM1 Controls the driving capability of BPI_BUS1.

- **0** The output driving capability is 2mA.
- **1** The output driving capability is 8mA.

PINM2 Controls the driving capability of BPI_BUS2.

- **0** The output driving capability is 2mA.
- 1 The output driving capability is 8mA.

0x82020004 BPI data register 0 BPI_BUF0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**PO9**	**PO8**	**PO7**	**PO6**	**PO5**	**PO4**	**PO3**	**PO2**	**PO1**	**PO0**
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register defines the BPI signals that are associated with the event TDMA_BPI0.

Table 71 lists 42 registers of the same structure, each of which is associated with one specific event signal from the TDMA timer. The data registers are all double-buffered. When PETEV is set to 0, the data register links to the write buffer. When PETEV is set to 1, the data register links to the active buffer.

One register, BPI_BUFI, is dedicated for use in immediate mode. Writing a value to that register effects an immediate change in the corresponding BPI signal and bus.

POx This flag defines the corresponding signals for BPIx after the TDMA event 0 takes place.

The overall data register definition is listed in **Table 71.**

Register Address	Register Function	Acronym

0x82020004	BPI pin data for event TDMA_BPI 0	BPI_BUF0
0x82020008	BPI pin data for event TDMA_BPI 1	BPI_BUF1
0x8202000C	BPI pin data for event TDMA_BPI 2	BPI_BUF2
0x82020010	BPI pin data for event TDMA_BPI 3	BPI_BUF3
0x82020014	BPI pin data for event TDMA_BPI 4	BPI_BUF4
0x82020018	BPI pin data for event TDMA_BPI 5	BPI_BUF5
0x8202001C	BPI pin data for event TDMA_BPI 6	BPI_BUF6
0x82020020	BPI pin data for event TDMA_BPI 7	BPI_BUF7
0x82020024	BPI pin data for event TDMA_BPI 8	BPI_BUF8
0x82020028	BPI pin data for event TDMA_BPI 9	BPI_BUF9
0x8202002C	BPI pin data for event TDMA_BPI 10	BPI_BUF10
0x82020030	BPI pin data for event TDMA_BPI 11	BPI_BUF11
0x82020034	BPI pin data for event TDMA_BPI 12	BPI_BUF12
0x82020038	BPI pin data for event TDMA_BPI 13	BPI_BUF13
0x8202003C	BPI pin data for event TDMA_BPI 14	BPI_BUF14
0x82020040	BPI pin data for event TDMA_BPI 15	BPI_BUF15
0x82020044	BPI pin data for event TDMA_BPI 16	BPI_BUF16
0x82020048	BPI pin data for event TDMA_BPI 17	BPI_BUF17
0x8202004C	BPI pin data for event TDMA_BPI 18	BPI_BUF18
0x82020050	BPI pin data for event TDMA_BPI 19	BPI_BUF19
0x82020054	BPI pin data for event TDMA_BPI 20	BPI_BUF20
0x82020058	BPI pin data for event TDMA_BPI 21	BPI_BUF21
0x8202005C	BPI pin data for event TDMA_BPI 22	BPI_BUF22
0x82020060	BPI pin data for event TDMA_BPI 23	BPI_BUF23
0x82020064	BPI pin data for event TDMA_BPI 24	BPI_BUF24
0x82020068	BPI pin data for event TDMA_BPI 25	BPI_BUF25
0x8202006C	BPI pin data for event TDMA_BPI 26	BPI_BUF26
0x82020070	BPI pin data for event TDMA_BPI 27	BPI_BUF27
0x82020074	BPI pin data for event TDMA_BPI 28	BPI_BUF28
0x82020078	BPI pin data for event TDMA_BPI 29	BPI_BUF29
0x8202007C	BPI pin data for event TDMA_BPI 30	BPI_BUF30
0x82020080	BPI pin data for event TDMA_BPI 31	BPI_BUF31
0x82020084	BPI pin data for event TDMA_BPI 32	BPI_BUF32
0x82020088	BPI pin data for event TDMA_BPI 33	BPI_BUF33
0x8202008C	BPI pin data for event TDMA_BPI 34	BPI_BUF34
0x82020090	BPI pin data for event TDMA_BPI 35	BPI_BUF35
0x82020094	BPI pin data for event TDMA_BPI 36	BPI_BUF36
0x82020098	BPI pin data for event TDMA_BPI 37	BPI_BUF37

© 2009 MediaTek Inc.

0x8202009C	BPI pin data for event TDMA_BPI 38	BPI_BUF38
0x820200A0	BPI pin data for event TDMA_BPI 39	BPI_BUF39
0x820200A4	BPI pin data for event TDMA_BPI 40	BPI_BUF40
0x820200A8	BPI pin data for event TDMA_BPI 41	BPI_BUF41
0x820200AC	BPI pin data for immediate mode	BPI_BUFI

Table 71 *BPI Data Registers.*

0x820200B0 BPI event enable register 0 BPI_ENA0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BEN15	BEN14	BEN13	BEN12	BEN11	BEN10	BEN9	BEN8	BEN7	BEN6	BEN5	BEN4	BEN3	BEN2	BEN1	BEN0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register enables the events that are signaled by the TDMA timer: by clearing a register bit, the corresponding event signal is ignored. After a hardware reset, all the enable bits default to 1 (enabled). Upon receiving a TDMA_EVTVAL pulse, all register bits are also set to 1 (enabled).

BENn This flag indicates whether event n signals are heeded or ignored.

- **0** Event n is disabled (ignored).
- **1** Event n is enabled.

0x820200B4 BPI event enable register 1 BPI_ENA1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BEN31	BEN30	BEN29	BEN28	BEN27	BEN26	BEN25	BEN24	BEN23	BEN22	BEN21	BEN20	BEN19	BEN18	BEN17	BEN16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register enables the events that are signaled by the TDMA timing generator: by clearing a register bit, the corresponding event signal is ignored. After a hardware reset, all the enable bits default to 1 (enabled). Upon receiving the TDMA_EVTVAL pulse, all register bits are also set to 1 (enabled).

BENn This flag indicates whether event n signals are heeded or ignored.

- **0** Event n is disabled (ignored)
- **1** Event n is enabled

0x820200B8 BPI event enable register 2 BPI_ENA2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							BEN41	BEN40	BEN39	BEN38	BEN37	BEN36	BEN35	BEN34	BEN33	BEN32
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset							1	1	1	1	1	1	1	1	1	1

The register is used to enable the events that are signaled by the TDMA timing generator. After hardware reset, all the enable bits defaults to be 1 (enabled). Upon receiving the TDMA_EVTVAL pulse, those bits are also set to 1 (enabled).

BENn The flag controls the function of event n.

- 0 The event n is disabled.
- 1 The event n is enabled.

4.4.3 Application Note

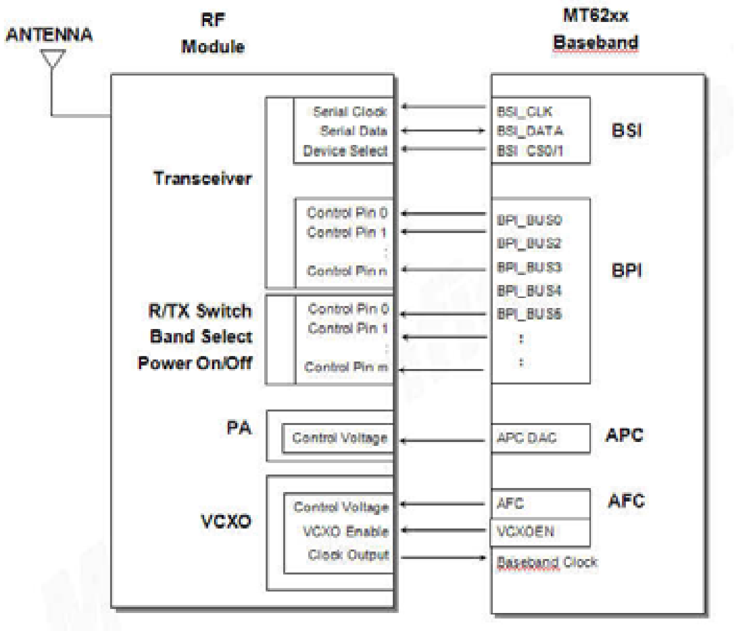


Figure 81 RF module connected with BPI

MT62xx also provides several GPO-like signals called BPI (Baseband Parallel Interface) Bus, which is activated at the specific time, to perform this control purpose. The timing of activating high/low states of signals on BPI bus is programmable, and the state of signals on BPI bus is changed on the specified timing. The timing of changing the states of BPI bus is called **BPI event**. The state of BPI signals with corresponding BPI event is call **BPI data**. **BPI event** can be programmable in the TDMA timer, and **BPI data** can be programmable in this module. In the above figure, BB can control R/Tx Switch, Band Select, and Power On/OFF etc. of RF module via BPI pins.

4.5 Baseband Serial Interface

The Baseband Serial Interface controls external radio components. A 3-wire serial bus transfers data to RF circuitry for PLL frequency change, reception gain setting, and other radio control purposes. In this unit, BSI data registers are double-buffered in the same way as the TDMA event registers. The user writes data into the write buffer and the data is transferred from the write buffer to the active buffer when a TDMA_EVTVAL signal (from the TDMA timer) is pulsed.

Each data register BSI_D*n*_DAT is associated with one data control register BSI_D*n*_CON, where *n* denotes the index. Each data control register identifies which events (signaled by TDMA_BSISTR*n*, generated by the TDMA timer) trigger the download process of the word in register BSI_D*n*_DAT. The word and its length (in bits) is downloaded via the serial bus. A special event is triggered when the IMOD flag is set to 1: it provides immediate download process without software programming the TDMA timer.

If more than one data word is to be downloaded on the same BSI event, the word with the lowest address among them is downloaded first, followed by the next lowest and so on.

 © 2009 MediaTek Inc.

The total download time depends on the word length, the number of words to download, and the clock rates. The programmer must space the successive event to provide enough time. If the download process of the previous event is not complete before a new event arrives, the latter is suppressed.

The unit has four output pins: BSI_CLK is the output clock, BSI_DATA is the serial data port, and BSI_CS0 and BSI_CS1 are the select pins for 2 external components. BSI_CS1 is multiplexed with another function. Please refer to GPIO table for more detail.

In order to support bi-directional read and write operations of the RF chip, software can directly write values to BSI_CLK, BSI_DATA and BSI_CS by programming the BSI_DOUT register. Data from the RF chip can be read by software via the register BSI_DIN. If the RF chip interface is a 3-wire interface, then BSI_DATA is bi-directional. Before software can program the 3-wire behavior, the BSI_IO_CON register must be set. An additional signal path from GPIO accommodates RF chips with a 4-wire interface.

The block diagram of the BSI unit is as depicted in **Figure 82.**

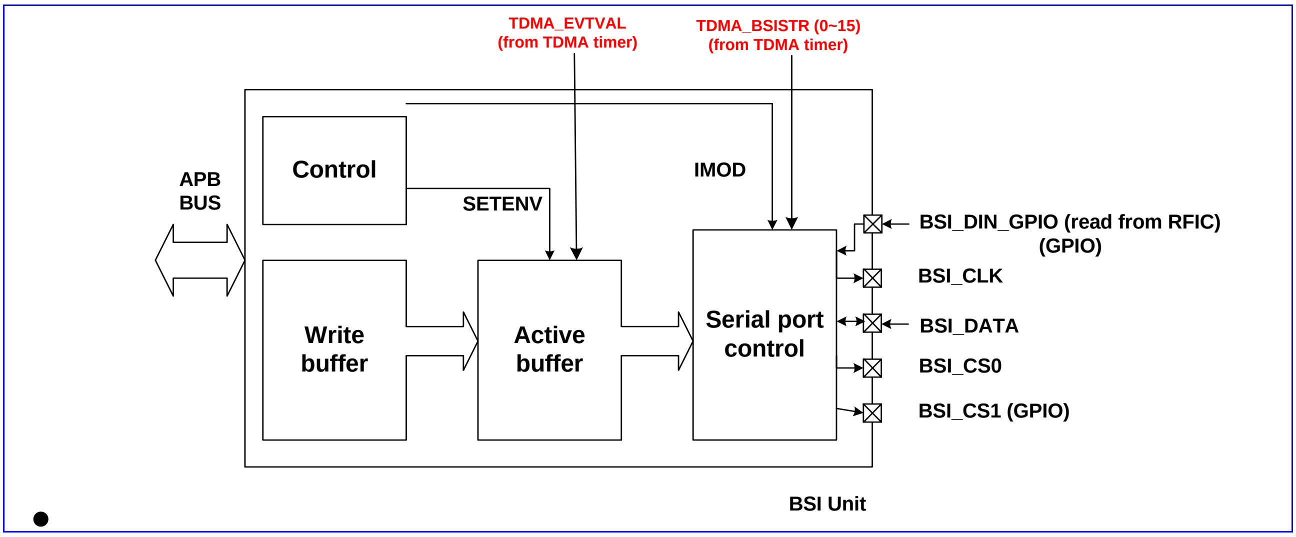


Figure 82 *Block diagram of BSI unit.*

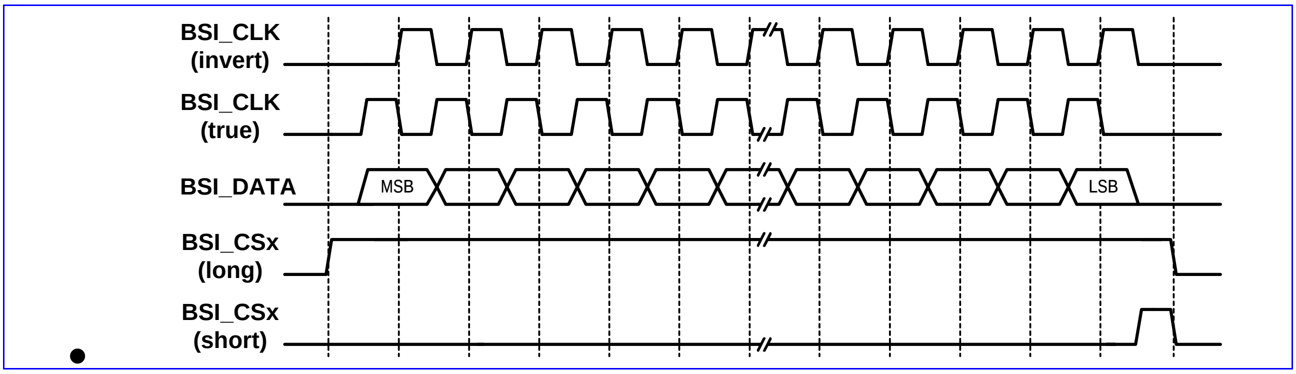


Figure 83 *Timing characteristic of BSI interface.*

© 2009 MediaTek Inc.

4.5.1 Register Definitions

Register Address	Register Function	Acronym
0x82010000	BSI control register	BSI_CON
0x82010190	BSI event enable register	BSI_ENA_0
0x82010194	BSI event enable register – MSB 4 bits	BSI_ENA_1
0x82010198	BSI IO mode control register	BSI_IO_CON
0x8201019C	Software-programmed data out	BSI_DOUT
0x820101A0	Input data from RF chip	BSI_DIN
0x820101A4	BSI data pair number	BSI_PAIR_NUM

Table 72 ***BSI Control Registers***

0x82010000 BSI control register BSI_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SETENV	EN1_POL	EN1_LEN	EN0_POL	EN0_LEN	IMOD	CLK_SPD		CLK_POL
Type								R/W	R/W	R/W	R/W	R/W	WO	R/W		R/W
Reset								0	0	0	0	0	N/A	0		0

This register is the control register for the BSI unit. The register controls the signal type of the 3-wire interface.

CLK_POL Controls the polarity of BSI_CLK. Refer to **Figure 83.**

- **0** True clock polarity
- 1 Inverted clock polarity

CLK_SPD Defines the clock rate of BSI_CLK. The 3-wire interface provides 4 choices of data bit rate. The default is 52/2 MHz.

- **00** 52/2 MHz
- **01** 52/4 MHz
- **10** 52/6 MHz
- **11** 52/8 MHz

IMOD Enables immediate mode. If the user writes 1 to the flag, the download is triggered immediately without waiting for the timer events. The words for which the register event ID equals 1Fh are downloaded following this signal. This flag is write-only. The immediate write is exercised only once: the programmer must write the flag again to invoke another immediate download. Setting the flag does not disable the other events from the timer; the programmer can disable all events by setting BSI_ENA to all zeros.

- **0** Reserved
- **1** Trigger Immediate Mode

ENX_LEN Controls the type of signals BSI_CS0 and BSI_CS1. Refer to **Figure 82.**

- **0** Long enable pulse
- 1 Short enable pulse

ENX_POL Controls the polarity of signals BSI_CS0 and BSI_CS1.

- **0** True enable pulse polarity

1 Inverted enable pulse polarity

SETENV Enables the write operation of the active buffer.

0 The user writes to the write buffer. The data is then latched in the active buffer after TDMA_EVTVAL is pulsed.

1 The user writes data directly to the active buffer.

0x82010004 Control part of data register 0 BSI_D0_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISB	LEN										EVT_ID				
Type	R/W	R/W										R/W				

This register is the control part of the data register 0. The register determines the required length of the download data word, the event to trigger the download process of the word, and the targeted device.

Table 74 lists the 44 data registers of this type. The max length of the first 40 data registers is 32 bits, and that of the last 4 data registers is 78 bits. Multiple data control registers may contain the same event ID. The data words of all registers with the same event ID are downloaded when the event occurs.

EVT_ID Stores the event ID for which the data word awaits to be downloaded.

00000~10011 Synchronous download of the word with the selected EVT_ID event. The relationship between this field and the event is listed as **Table 77.**

Event ID (in binary) – EVT_ID	Event name
00000	TDMA_BSISTR0
00001	TDMA_BSISTR1
00010	TDMA_BSISTR2
00011	TDMA_BSISTR3
00100	TDMA_BSISTR4
00101	TDMA_BSISTR5
00110	TDMA_BSISTR6
00111	TDMA_BSISTR7
01000	TDMA_BSISTR8
01001	TDMA_BSISTR9
01010	TDMA_BSISTR10
01011	TDMA_BSISTR11
01100	TDMA_BSISTR12
01101	TDMA_BSISTR13
01110	TDMA_BSISTR14
01111	TDMA_BSISTR15
10000	TDMA_BSISTR16
10001	TDMA_BSISTR17
10010	TDMA_BSISTR18
10011	TDMA_BSISTR19

Table 77 The relationship between the value of EVT_ID field in the BSI control registers and the TDMA_BSISTR events.

10100~11110 Reserved

11111 Immediate download

LEN The field stores the length of the data word. The actual length is defined as LEN + 1 in units of bits. For data registers 0~39, the value ranges from 0 to 31, corresponding to 1 to 32 bits in length. For data registers 40~43, the value ranges from 0 to 77, corresponding to 1 to 78 bits in length.

ISB The flag selects the target device.

0 Device 0 is selected.

1 Device 1 is selected.

0x82010008 Data part of data register 0 BSI_D0_DAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT [31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT [15:0]															
Type	R/W															

This register is the data part of the data register 0. The legal length of the data is up to 32 bits. The actual number of bits to be transmitted is specified in LEN field in the BSI_D0_CON register.

DAT The field signifies the data part of the data register.

Table 74 lists the address mapping and function of the 44 pairs of data registers.

Register Address	Register Function	Acronym
0x82010004	Control part of data register 0	BSI_D0_CON
0x82010008	Data part of data register 0	BSI_D0_DAT
0x8201000C	Control part of data register 1	BSI_D1_CON
0x82010010	Data part of data register 1	BSI_D1_ DAT
0x82010014	Control part of data register 2	BSI_D2_CON
0x82010018	Data part of data register 2	BSI_D2_ DAT
0x8201001C	Control part of data register 3	BSI_D3_CON
0x82010020	Data part of data register 3	BSI_D3_ DAT
0x82010024	Control part of data register 4	BSI_D4_CON
0x82010028	Data part of data register 4	BSI_D4_ DAT
0x8201002C	Control part of data register 5	BSI_D5_CON
0x82010030	Data part of data register 5	BSI_D5_ DAT
0x82010034	Control part of data register 6	BSI_D6_CON
0x82010038	Data part of data register 6	BSI_D6_ DAT
0x8201003C	Control part of data register 7	BSI_D7_CON
0x82010040	Data part of data register 7	BSI_D7_ DAT
0x82010044	Control part of data register 8	BSI_D8_CON

 © 2009 MediaTek Inc.

0x82010048	Data part of data register 8	BSI_D8_ DAT
0x8201004C	Control part of data register 9	BSI_D9_CON
0x82010050	Data part of data register 9	BSI_D9_ DAT
0x82010054	Control part of data register 10	BSI_D10_CON
0x82010058	Data part of data register 10	BSI_D10_ DATA
0x8201005C	Control part of data register 11	BSI_D11_CON
0x82010060	Data part of data register 11	BSI_D11_ DAT
0x82010064	Control part of data register 12	BSI_D12_CON
0x82010068	Data part of data register 12	BSI_D12_ DAT
0x8201006C	Control part of data register 13	BSI_D13_CON
0x82010070	Data part of data register 13	BSI_D13_ DAT
0x82010074	Control part of data register 14	BSI_D14_CON
0x82010078	Data part of data register 14	BSI_D14_ DAT
0x8201007C	Control part of data register 15	BSI_D15_CON
0x82010080	Data part of data register 15	BSI_D15_ DAT
0x82010084	Control part of data register 16	BSI_D16_CON
0x82010088	Data part of data register 16	BSI_D16_ DAT
0x8201008C	Control part of data register 17	BSI_D17_CON
0x82010090	Data part of data register 17	BSI_D17_ DAT
0x82010094	Control part of data register 18	BSI_D18_CON
0x82010098	Data part of data register 18	BSI_D18_ DAT
0x8201009C	Control part of data register 19	BSI_D19_CON
0x820100A0	Data part of data register 19	BSI_D19_ DAT
0x820100A4	Control part of data register 20	BSI_D20_CON
0x820100A8	Data part of data register 20	BSI_D20_ DAT
0x820100AC	Control part of data register 21	BSI_D21_CON
0x820100B0	Data part of data register 21	BSI_D21_ DAT
0x820100B4	Control part of data register 22	BSI_D22_CON
0x820100B8	Data part of data register 22	BSI_D22_ DAT
0x820100BC	Control part of data register 23	BSI_D23_CON
0x820100C0	Data part of data register 23	BSI_D23_ DAT
0x820100C4	Control part of data register 24	BSI_D24_CON
0x820100C8	Data part of data register 24	BSI_D24_ DAT
0x820100CC	Control part of data register 25	BSI_D25_CON
0x820100D0	Data part of data register 25	BSI_D25_ DAT
0x820100D4	Control part of data register 26	BSI_D26_CON
0x820100D8	Data part of data register 26	BSI_D26_ DAT
0x820100DC	Control part of data register 27	BSI_D27_CON

0x820100E0	Data part of data register 27	BSI_D27_ DAT
0x820100E4	Control part of data register 28	BSI_D28_CON
0x820100E8	Data part of data register 28	BSI_D28_ DAT
0x820100EC	Control part of data register 29	BSI_D29_CON
0x820100F0	Data part of data register 29	BSI_D29_ DAT
0x820100F4	Control part of data register 30	BSI_D30_CON
0x820100F8	Data part of data register 30	BSI_D30_ DAT
0x820100FC	Control part of data register 31	BSI_D31_CON
0x82010100	Data part of data register 31	BSI_D31_ DAT
0x82010104	Control part of data register 32	BSI_D32_CON
0x82010108	Data part of data register 32	BSI_D32_ DAT
0x8201010C	Control part of data register 33	BSI_D33_CON
0x82010110	Data part of data register 33	BSI_D33_ DAT
0x82010114	Control part of data register 34	BSI_D34_CON
0x82010118	Data part of data register 34	BSI_D34_ DAT
0x8201011C	Control part of data register 35	BSI_D35_CON
0x82010120	Data part of data register 35	BSI_D35_ DAT
0x82010124	Control part of data register 36	BSI_D36_CON
0x82010128	Data part of data register 36	BSI_D36_ DAT
0x8201012C	Control part of data register 37	BSI_D37_CON
0x82010130	Data part of data register 37	BSI_D37_ DAT
0x82010134	Control part of data register 38	BSI_D38_CON
0x82010138	Data part of data register 38	BSI_D38_ DAT
0x8201013C	Control part of data register 39	BSI_D39_CON
0x82010140	Data part of data register 39	BSI_D39_ DAT
0x82010144	Control part of data register 40	BSI_D40_CON
0x82010148	Data part of data register 40 (MSB 14 bits)	BSI_D40_ DAT2
0x8201014C	Data part of data register 40	BSI_D40_ DAT1
0x82010150	Data part of data register 40 (LSB 32 bits)	BSI_D40_ DAT0
0x82010154	Control part of data register 41	BSI_D41_CON
0x82010158	Data part of data register 41 (MSB 14 bits)	BSI_D41_ DAT2
0x8201015C	Data part of data register 41	BSI_D41_ DAT1
0x82010160	Data part of data register 41 (LSB 32 bits)	BSI_D41_ DAT0
0x82010164	Control part of data register 42	BSI_D42_CON
0x82010168	Data part of data register 42 (MSB 14 bits)	BSI_D42_ DAT2
0x8201016C	Data part of data register 42	BSI_D42_ DAT1
0x82010170	Data part of data register 42 (LSB 32 bits)	BSI_D42_ DAT0
0x82010174	Control part of data register 43	BSI_D43_CON

 © 2009 MediaTek Inc.

0x82010178	Data part of data register 43 (MSB 14 bits)	BSI_D43_ DAT2
0x8201017C	Data part of data register 43	BSI_D43_ DAT1
0x82010180	Data part of data register 43 (LSB 32 bits)	BSI_D43_ DAT0

Table 74 *BSI data registers*

0x82010190 BSI event enable register BSI_ENA_0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSI15	BSI14	BSI13	BSI12	BSI11	BSI10	BSI9	BSI8	BSI7	BSI6	BSI5	BSI4	BSI3	BSI2	BSI1	BSI0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register enables an event by setting the corresponding bit. After a hardware reset, all bits are initialized to 1. These bits are also set to 1 after TDMA_EVTVAL pulse.

BSIx Enables downloading of the words corresponding to the events signaled by TMDA_BSI.

- **0** The event is not enabled.
- 1 The event is enabled.

0x82010194 BSI event enable register – MSB 4 bits BSI_ENA_1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BSI19	BSI18	BSI17	BSI16
Type													R/W	R/W	R/W	R/W
Reset													1	1	1	1

The register could enable the event by setting the corresponding bit. After hardware reset, all bits are initialized as 1. Besides, those bits are set as 1 after TDMA_EVTVAL is pulsed.

BSIx The flag enables the downloading of the words that corresponds to the events signaled by TMDA_BSI.

- 0 The event is not enabled.
- The event is enabled.

0x82010198 BSI IO mode control register BSI_IO_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SEL_CS1	4_WIRE	DAT_DIR	MODE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

MODE Defines the source of BSI signal.

- 0 BSI signal is generated by the hardware.
- 1 BSI signal is generated by the software. In this mode, the BSI clock depends on the value of the field DOUT.CLK. BSI_CS depends on the value of the field DOUT.CS and BSI_DATA depends on the value of the field DOUT.DATA.

DAT_DIR Defines the direction of BSI_DATA.

- 0 BSI _DATA is configured as input. The 3-wire interface is used and BSI_DATA is bi-directional.
- 1 BSI_DATA is configured as output.

4_WIRE Defines the BSI_DIN source.

0 The 3-wire interface is used and BSI_DATA is bi-directional. BSI_DIN comes from the same pin as BSI_DATA.

1 The 4-wire interface is used. Another pin (GPIO) is used as BSI_DIN.

SEL_CS1 Defines which of the BSI_CSx (BSI_CS0 or BSI_CS1) is written by the software.

0 BSI_CS0 is selected.

1 BSI_CS1 is selected.

0x8201019C Software-programmed data out BSI_DOUT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														DATA	CS	CLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CLK Signifies the BSI_CLK signal.

CS Signifies the BSI_CS signal.

DATA Signifies the BSI_DATA signal.

0x820101A0 Input data from RF chip BSI_DIN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DIN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIN Registers the input value of BSI_DATA from the RF chip.

0x820101A4 BSI data pair number BSI_PAIR_NUM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PAIR_NUM					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R					
Reset	0	0	0	0	0	0	0	0	0	0	28					

PAIR_NUM The software can program how many pairs of data register to be used. The default value is 28 pairs. This value must be smaller or equal to 44. The first 40 pairs are 32-bit long, and the last four pairs are 78-bit long.

 © 2009 MediaTek Inc.

4.5.2 Application Note

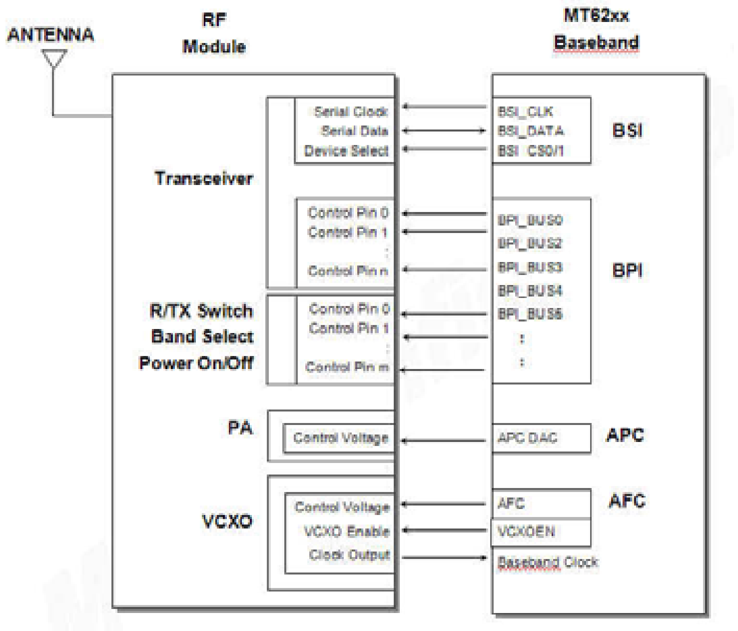


Figure 84 RF module connected with BSI (3-wire) interface

Some RF chips can be commanded by receiving serial signals. 3 wires, i.e. device select, serial clock, and serial data, are needed to control the device. In MT62xx baseband chip, this interface named **BSI** (Baseband Serial Interface) is provided. The timing to activate BSI unit sending 3-wire control signals is called **BSI event**. The serial data sent with corresponding BSI event is call **BSI data**. **BSI event** can be programmable in the TDMA module, and **BSI data** can be programmable at this module. BSI module can support different clock rates and device select types. The output clock rate depends on RF module's spec & SW's timing budget. Because BSI is serial transmission, it needs some time to send the whole data words. It must be considered in the SW's timing budget.

4.6 CSD Accelerator

4.6.1 General Description

This unit performs the data format conversion of RA0, RA1, and FAX in CSD service. CSD service consists of two major functions: data flow throttling and data format conversion. The data format conversion is a bit-wise operation and takes a number of instructions to complete a conversion. Therefore, it is not efficient to do by MCU itself. A coprocessor, CSD accelerator, is designed here to reduce the computing power needed to perform this function.

CSD accelerator only helps in converting data format; the data flow throttling function is still implemented by the MCU. CSD accelerator performs three types of data format conversion, RA0, RA1, and FAX.

For RA0 conversion, only uplink RA0 data format conversion is provided here. This is because there are too many judgments on the downlink path conversion, which will greatly increase area cost. Uplink RA0

 © 2009 MediaTek Inc.

conversion is to insert one start bit and one stop bit before and after a byte, respectively, during 16 bytes. **Figure 85** illustrates the detailed conversion table.

RA0 converter can only process RA0 data state by state. Before filling in new data, software must make sure the converted data of certain state is withdrawn, or the converted data will be replaced by the new data. For example, if 32-bit data is written, and the state pointer goes from state 0 to state 1, and word ready of state 0 is asserted; then, before writing the next 32-bit data, the word of state 0 should be withdrawn first, or the data will be lost.

RA0 records the number of written bytes, state pointer, and ready state word. The information can help software to perform flow control. See Register Definition for more detail.

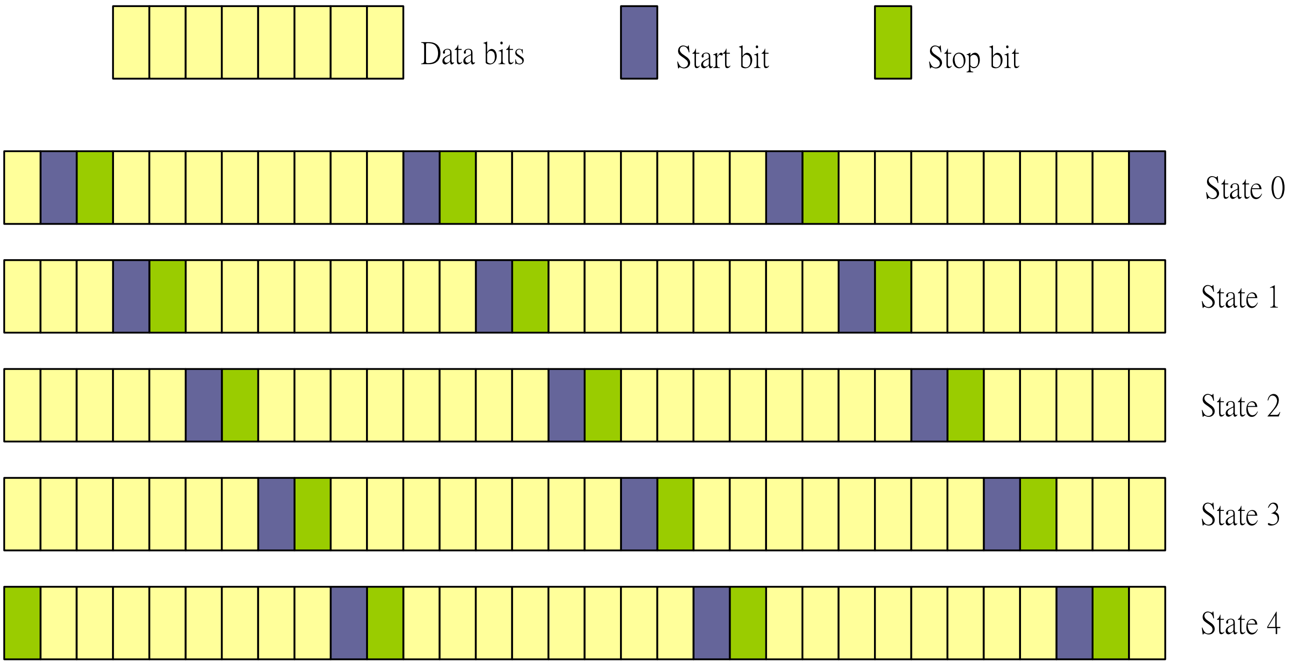


Figure 85 ***data format conversion of RA0***

For RA1 conversion, both directions, downlink and uplink, are supported. The data formats vary in different data rate. The detailed conversion table is shown in **Figure 86** and **Figure 87**. The yellow part is the payload data, and the blue part is the status bit.

© 2009 MediaTek Inc.

Bit 0 → Bit 6

D1	D2	D3	D4	D5	D6	S1
D7	D8	D9	D10	D11	D12	X
D13	D14	D15	D16	D17	D18	S3
D19	D20	D21	D22	D23	D24	S4
E4	E5	E6	E7	D25	D26	D27
D28	D29	D30	S6	D31	D32	D33
D34	D35	D36	X	D37	D38	D39
D40	D41	D42	S8	D43	D44	D45
D46	D47	D48	S9			

Bit 59

Figure 86 ***data format conversion for 6k/12k RA1***

Bit 0 → Bit 7

D1	D2	D3	S1	D4	D5	D6	X
D7	D8	D9	S3	D10	D11	D12	S4
E4	E5	E6	E7	D13	D14	D15	S6
D16	D17	D18	X	D19	D20	D21	S8
D22	D23	D24	S9				

Bit 35

Figure 87 ***data format conversion for 3.6k RA1***

For FAX, two types of bit-reversal functions are provided. One is bit-wise reversal, and the other is byte-wise reversal, which are illustrated in **Figure 88** and **Figure 89**, respectively.

© 2009 MediaTek Inc.

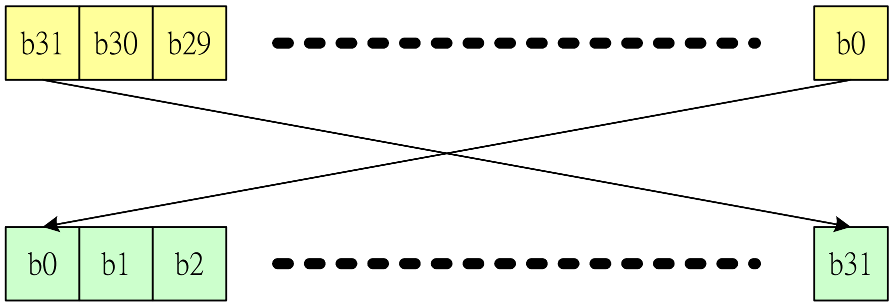


Figure 88 ***Type 1 bit reverse***

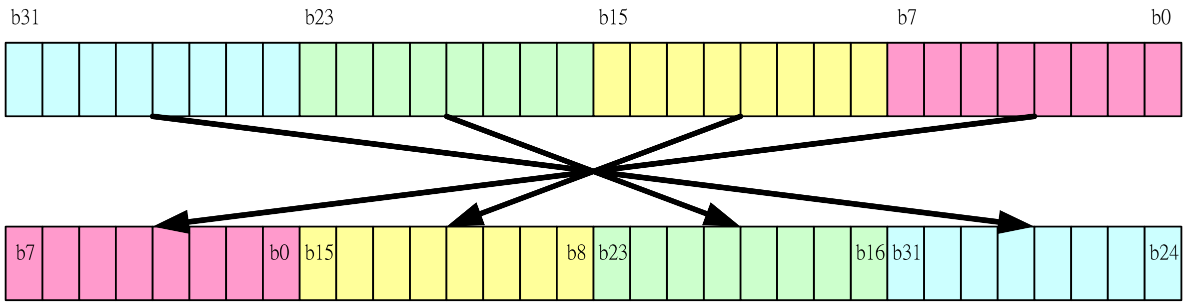


Figure 89 ***Type 2 bit reverse***

Register Address	**Register Function**	**Acronym**
0x82090000	CSD RA0 Control Register	**CSD_RA0_CON**
0x82090004	CSD RA0 Status Register	**CSD_RA0_STA**
0x82090008	CSD RA0 Input Data Register	**CSD_RA0_DI**
0x8209000C	CSD RA0 Output Data Register	**CSD_RA0_DO**
0x82090100	CSD RA1 6K/12K Uplink Input Data Register 0	**CSD_RA1_6K_12K_ULDI0**
0x82090104	CSD RA1 6K/12K Uplink Input Data Register 1	**CSD_RA1_6K_12K_ULDI1**
0x82090108	CSD RA1 6K/12K Uplink Status Data Register	**CSD_RA1_6K_12K_ULSTUS**
0x8209010C	CSD RA1 6K/12K Uplink Output Data Register 0	**CSD_RA1_6K_12K_ULDO0**
0x82090110	CSD RA1 6K/12K Uplink Output Data Register 1	**CSD_RA1_6K_12K_ULDO1**
0x82090200	CSD RA1 6K/12K Downlink Input Data Register 0	**CSD_RA1_6K_12K_DLDI0**
0x82090204	CSD RA1 6K/12K Downlink Input Data Register 1	**CSD_RA1_6K_12K_DLDI1**
0x82090208	CSD RA1 6K/12K Downlink Output Data Register 0	**CSD_RA1_6K_12K_DLDO0**
0x8209020C	CSD RA1 6K/12K Downlink Output Data Register 1	**CSD_RA1_6K_12K_DLDO1**
0x82090210	CSD RA1 6K/12K Downlink Status Data Register	**CSD_RA1_6K_12K_DLSTUS**
0x82090300	CSD RA13.6K Uplink Input Data Register 0	**CSD_RA1_3P6K_ULDI0**
0x82090304	CSD RA13.6K Uplink Status Data Register	**CSD_RA1_3P6K_ULSTUS**
0x82090308	CSD RA13.6K Uplink Output Data Register 0	**CSD_RA1_3P6K_ULDO0**
0x8209030C	CSD RA13.6K Uplink Output Data Register 1	**CSD_RA1_3P6K_ULDO1**

 © 2009 MediaTek Inc.

0x82090400	CSD RA1 3.6K Downlink Input Data Register 0	**CSD_RA1_3P6K_DLDI0**
0x82090404	CSD RA1 3.6K Downlink Input Data Register 1	**CSD_RA1_3P6K_DLDI1**
0x82090408	CSD RA1 3.6K Downlink Output Data Register 0	**CSD_RA1_3P6K_DLDO0**
0x8209040C	CSD RA1 3.6K Downlink Status Data Register	**CSD_RA1_3P6K_DLSTUS**
0x82090500	CSD FAX Bit Reverse Type 1 Input Data Register	**CSD_FAX_BR1_DI**
0x82090504	CSD FAX Bit Reverse Type 1 Output Data Register	**CSD_FAX_BR1_DO**
0x82090510	CSD FAX Bit Reverse Type 2 Input Data Register	**CSD_FAX_BR2_DI**
0x82090514	CSD FAX Bit Reverse Type 2 Output Data Register	**CSD_FAX_BR2_DO**

Table 79 ***CSD Accelerater Registers***

4.6.2 Register Definitions

0X82090000 CSD RA0 Control Register CSD_RA0_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name											**RST**	**BTS0**		**VLD_BYTE**		
Type											WO	WO		WO		
Reset											0	0		100		

VLD_BYTE Specify how many valid bytes in the current input data. It must be specified before filling data in.

BTS0 Back to state 0. Force RA0 converter go back to state 0. Incomplete word will be padded by STOP bit. For instance, back-to-state0 command is issued after 8 byte data are filled in. Then these bit after the 8th byte will be padded with stop bits, and RDYWD2 is asserted. After removing state word 2, the state pointer goes back to state 0. Note that new data filling should take place after removing state word 2, or the state pointer may be out of order.

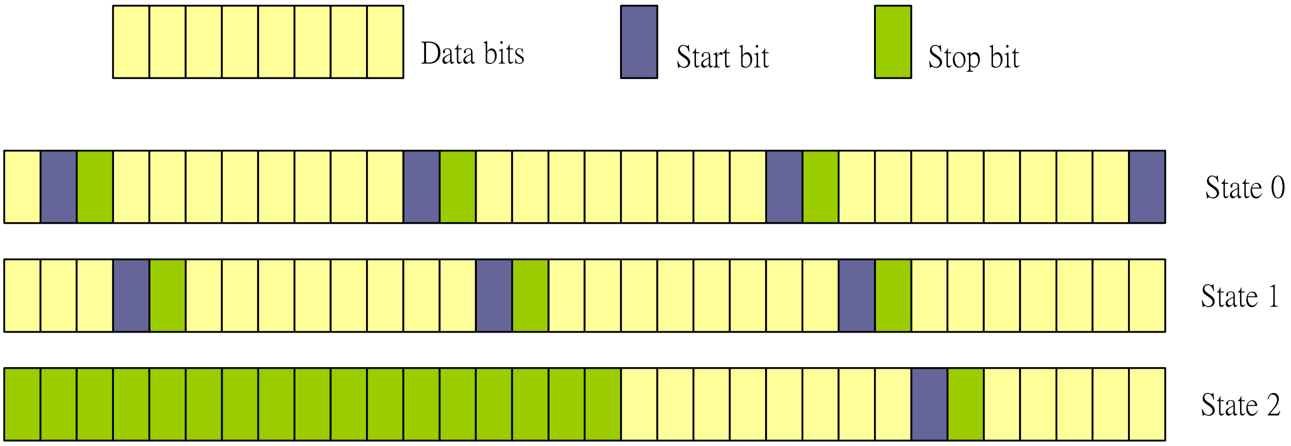


Figure 90 ***Example of Back to state 0***

© 2009 MediaTek Inc.

RST Reset RA0 converter. In case, erroneously operation makes data disordered. This bit can restore all state to original state.

- **0** Reserved
- **1** Reset

0X82090004 CSD RA0 Status Register CSD_RA0_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**BYTECNT**				**CRTSTA**			**RDYWD**				
Type					RO				RO			RC				
Reset					0				0			0				

RDYWD0~4 Ready word. To indicate which state word is ready for withdrawal. Data should be withdrawn before next data fills into CSD_RA0_DI, if there are any bits asserted.

- **0** Not ready
- **1** Ready

CRTSTA Current state. State0 ~ State4. To indicate which state word software is filling in.

BYTECNT The total number of bytes software is filling in.

0X82090008 CSD RA0 Input Data Register CSD_RA0_DI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DIN**															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIN**															
Type	WO															
Reset	0															

DIN The RA0 convert input data. Ready word indicator shall be check before filling in data. If any words are ready, withdraw them first; otherwise the ready data in RA0 converter will be replaced.

0X8209000C CSD RA0 Output Data Register CSD_RA0_DO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DOUT**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DOUT**															
Type	RO															
Reset	0															

DOUT RA0 converted data. The return data corresponds to the ready word indicator defined in CSD_RA0_STA register. The five bit of RDYWD map to state0 ~ state 4 accordingly. When CSD_RA0_DO is read, the asserted state word will be returned. If there are two state words asserted at the same time, the lower one will be returned.

0X82090100 CSD RA1 6K/12K Uplink Input Data Register 0 CSD_RA1_6K_12K_ULDI0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIN															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															
Reset	0															

DIN The D1 to D32 of RA1 uplink data.

0X82090104 CSD RA1 6K/12K Uplink Input Data Register 1 CSD_RA1_6K_12K_ULDI1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															
Reset	0															

DIN The D33 to D48 of RA1 uplink data.

0X82090108 CSD RA1 6K/12K Uplink Status Data Register CSD_RA1_6K_12K_ULSTUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										E7	E6	E5	E4	X	SB	SA
Type										WO	WO	WO	WO	WO	WO	WO
Reset										0	0	0	0	0	0	0

SA Represents S1, S3, S6, and S8 of status bits.
SB Represents S4 and S9 of status bits.
X Represents X of status bits.
E4 Represents E4 of status bits.
E5 Represents E5 of status bits.
E6 Represents E6 of status bits.
E7 Represents E7 of status bits.

0X8209010C CSD RA1 6K/12K Uplink Output Data Register 0 CSD_RA1_6K_12K_ULDO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DOUT															

Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DOU**															
Type	RO															
Reset	0															

DOUT The bit 0 to bit 31 of RA1 6K/12K uplink frame.

0X82090110 CSD RA1 6K/12K Uplink Output Data Register 1 CSD_RA1_6K_12K_ULDO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**DOUT**											
Type					RO											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DOUT**															
Type	RO															
Reset	0															

DOUT The bit32 to bit 59 of RA1 6K/12K uplink frame.

0X82090200 CSD RA1 6K/12K Downlink Input Data Register 0 CSD_RA1_6K_12K_DLDI0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DIN**															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIN**															
Type	WO															
Reset	0															

DIN The bit 0 to bit 31 of RA1 6K/12K downlink frame.

0X82090204 CSD RA1 6K/12K Downlink Input Data Register 1 CSD_RA1_6K_12K_DLDI1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**DIN**											
Type					WO											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIN**															
Type	WO															
Reset	0															

DIN The bit32 to bit 59 of RA1 6K/12K downlink frame.

0X82090208 CSD RA1 6K/12K Downlink Output Data Register 0 CSD_RA1_6K_12K_DLDO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	DOUT															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DOUT															
Type	RO															
Reset	0															

DOUT The D1 to D32 of RA1 downlink data.

0X8209020C CSD RA1 6K/12K Downlink Output Data Register 1 CSD_RA1_6K_12K_DLDO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DOUT															
Type	RO															
Reset	0															

DOUT The D33 to D48 of RA1 downlink data.

0X82090210 CSD RA1 6K/12K Downlink Status Data Register CSD_RA1_6K_12K_DLSTUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										E7	E6	E5	E4	X	SB	SA
Type										RO	RO	RO	RO	RO	RO	RO
Reset										0	0	0	0	0	0	0

SA The result of majority votes of S1, S3, S6 and S8. SA is “0” if equal vote.

SB The result of majority votes of S4 and S9. SB is “0” if equal vote.

X The result of majority votes of two X bits in downlink frame. X is “0” if equal vote.

E4 Represents E4 of status bits.

E5 Represents E5 of status bits.

E6 Represents E6 of status bits.

E7 Represents E7 of status bits.

0X82090300 CSD RA1 3.6K Uplink Input Data Register 0 CSD_RA1_3P6K_ULDI0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									DIN							
Type									WO							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															

Reset	0

DIN The D1 to D24 of RA1 3.6K uplink data.

0X82090304 CSD RA1 3.6K Uplink Status Data Register CSD_RA1_3P6K_ULSTUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**E7**	**E6**	**E5**	**E4**	**X**	**SB**	**SA**
Type										WO	WO	WO	WO	WO	WO	WO
Reset										0	0	0	0	0	0	0

SA Represents S1, S3, S6, and S8 of status bits.
SB Represents S4 and S9 of status bits.
X Represents X of status bits.
E4 Represents E4 of status bits.
E5 Represents E5 of status bits.
E6 Represents E6 of status bits.
E7 Represents E7 of status bits.

0X82090308 CSD RA1 3.6K Uplink Output Data Register 0 CSD_RA1_3P6K_ULDO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DOUT**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DOUT**															
Type	RO															
Reset	0															

DOUT The bit 0 to bit 31 of RA1 3.6K uplink frame

0X8209030C CSD RA1 3.6K Uplink Output Data Register 1 CSD_RA1_3P6K_ULDO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**DOUT**			
Type													RO			
Reset													0			

DOUT The bit 32 to bit 35 of RA1 3.6K uplink frame

0X82090400 CSD RA1 3.6K Downlink Input Data Register 0 CSD_RA1_3P6K_DLDI0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIN															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															
Reset	0															

DIN The bit 0 to bit 31 of RA1 3.6K downlink frame

0X82090404 CSD RA1 3.6K Downlink Input Data Register 1 CSD_RA1_3P6K_DLDI1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													DIN			
Type													WO			
Reset													0			

DIN The bit 32 to bit 35 of RA1 3.6K downlink frame

0X82090408 CSD RA1 3.6K Downlink Output Data Register 0 CSD_RA1_3P6K_DLDO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									DOUT							
Type									RO							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DOUT															
Type	RO															
Reset	0															

DIN The D1 to D24 of RA1 3.6K downlink data.

0X8209040C CSD RA1 3.6K Downlink Status Data Register CSD_RA1_3P6K_DLSTUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										E7	E6	E5	E4	X	SB	SA
Type										RO	RO	RO	RO	RO	RO	RO
Reset										0	0	0	0	0	0	0

SA The result of majority votes of S1, S3, S6 and S8. SA is “0” if equal vote.

SB The result of majority votes of S4 and S9. SB is “0” if equal vote.
X The result of majority votes of two X bits in downlink frame. X is “0” if equal vote.
E4 Represents E4 of status bits.
E5 Represents E5 of status bits.
E6 Represents E6 of status bits.
E7 Represents E7 of status bits.

0X82090500 CSD FAX Bit Reverse Type 1 Input Data Register CSD_FAX_BR1_DI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIN															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															
Reset	0															

DIN 32-bit input data for type 1 bit reverse of FAX data. The action of Type 1 bit reverse is to reverse this word by word.

0X82090504 CSD FAX Bit Reverse Type 1 Output Data Register CSD_FAX_BR1_DO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DOUT															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DOUT															
Type	RO															
Reset	0															

DOUT 32-bit result data for type 1 bit reverse of FAX data.

0X82090510 CSD FAX Bit Reverse Type 2 Input Data Register CSD_FAX_BR2_DI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIN															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DIN															
Type	WO															
Reset	0															

DIN 32-bit input data for type 2 bit reverse of FAX data. The action of Type 1 bit reverse is to reverse this word by byte.

0X82090514 CSD FAX Bit Reverse Type 2 Output Data Register CSD_FAX_BR2_DO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DOUT															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DOUT															
Type	RO															
Reset	0															

DOUT 32-bit result data for type 2 bit reverse of FAX data.

4.7 Divider

To ease the processing load of MCU, a divider is employed here. The divider can operate signed and unsigned 32bit/32bit division, as well as modulus. The processing time of the divider is from 1 clock cycle to 33 clock cycles, which depends upon the magnitude of the value of the dividend. The detailed processing time is listed below in **Table 6**. From the table we can see that there are two kind of processing time (except for when the dividend is zero) in an item. Which kind depends on whether there is the need for restoration at the last step of the division operation.

After the divider is started by setting START to “1” in Divider Control Register, DIV_RDY will go low, and it will be asserted after the division process is finished. MCU could detect this status bit by polling it to know the correct access timing. In order to simplify polling, only the value of register DIV_RDY will appear while Divider Control Register is read. Hence, MCU does not need to mask other bits to extract the value of DIV_RDY.

In GSM/GPRS system, many divisions are executed with some constant divisors. Therefore, some often-used constants are stored in the divider to speed up the process. By controlling control bits IS_CNST and CNST_IDX in Divider Control register, one can start a division without giving a divisor. This could save the time for writing divisor in and the instruction fetch time, and thus make the process more efficient.

Signed Division		Unsigned Division	
Dividend	**Clock Cycles**	**Dividend**	**Clock Cycles**
0000_0000h	1	0000_0000h	1
0000_00ffh – (-0000_0100h), excluding 0x0000_0000	8 or 9	0000_0001h - 0000_00ffh	8 or 9
0000_ffffh – (-0001_0000h)	16 or 17	0000_0100h - 0000_ffffh	16 or 17
00ff_ffffh – (-0100_0000h)	24 or 25	0001_0000h - 00ff_ffffh	24 or 25
7fff_ffffh – (-8000_0000h)	32 or 33	0100_0000h - ffff_ffffh	32 or 33

Table 80 ***Processing time in different value of dividend.***

4.7.1 Register Definitions

Register Address	Register Function	Acronym
0X82060000	Divider Control Register	DIV_CON

0X82060004	Divider Dividend register	DIV_DIVIDEND
0X82060008	Divider Divisor register	DIV_DIVISOR
0X8206000C	Divider Quotient register	DIV_QUOTIENT
0X82060010	Divider Remainder register	DIV_REMAINDER

Table 81 ***All Registers Table***

0X82060000 Divider Control Register DIV_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														**CNST_IDX**		
Type														WO		
Reset														0		
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name											**IN_CNST**	**SIGN**			**DIV_RDY**	**START**
Type											WO	WO			RO	WO
Reset											0	1			1	0

START To start division. It will return to 0 after division has started.

- **0** Reserved
- **1** Start division

DIV_RDY Current status of divider. Note that when DIV_CON register is read, only the value of DIV_RDY will appear. That means program does not need to mask other part of the register to extract the information of DIV_RDY.

- **0** division is in progress.
- 1 division is finished.

SIGN To indicate signed or unsigned division.

- **0** Unsigned division.
- **1** Signed division.

IS_CNST To indicate if internal constant value should be used as a divisor. If IS_CNST is enabled, User does not need to write the value of the divisor, and divider will automatically use the internal constant value instead. What value divider will use depends on the value of CNST_IDX.

- **0** Normal division. Divisor is written in via APB
- **1** Using internal constant divisor instead.

CNST_IDX Index of constant divisor.

- **0** divisor = 13
- 1 divisor = 26
- 2 divisor = 51
- 3 divisor = 52
- 4 divisor = 102
- 5 divisor = 104

0X82060004 Divider Dividend register DIV_DIVIDEND

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DIVIDEND[31:16]**															
Type	WO															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIVIDEND[15:0]**															
Type	WO															
Reset	0															

DIVIDEND Dividend

0X82060008 Divider Divisor register DIV_DIVISOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DIVISOR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DIVISOR[15:0]**															
Type	R/W															
Reset	0															

Divisor Divisor

0X8206000C Divider Quotient register DIV_QUOTIENT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**QUOTIENT[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**QUOTIENT[15:0]**															
Type	RO															
Reset	0															

QUOTIENT Quotient

0X82060010 Divider Remainder register DIV_REMAINDER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**REMAINDER[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**REMAINDER[15:0]**															
Type	RO															
Reset	0															

REMAINDER Remainder

4.8 FCS Codec

4.8.1 General Description

FCS (Frame Check Sequence) is used to detect errors in the following information bits:

- RLP-frame of CSD services in GSM. The frame length is fixed as 240 or 576 bits including the 24-bit FCS field.

© 2009 MediaTek Inc.

- LLC-frame of GPRS service. The frame length is determined by the information field, and length of the FCS field is 24-bit.

Generation of the frame check sequence is very similar to the CRC coding in baseband signal processing. ETSI GSM specifications 04.22 and 04.64 both define the coding rule. The coding rules are:

1. The CRC shall be ones complement of the modulo-2 sum of:

- the remainder of $x^k \cdot (x^{23}+x^{22}+x^{21}+\ldots+x^2+x+1)$ modulo-2 divided by the generator polynomial, where k is the number of bits of the dividend. (i.e. fill the shift registers with all ones initially before feeding data)
- the remainder of the modulo-2 division by the generator polynomial of the product of x^{24} by the dividend, which are the information bits.

2. The CRC-24 generator polynomial is:

$G(x)=x^{24}+x^{23}+x^{21}+x^{20}+x^{19}+x^{17}+x^{16}+x^{15}+x^{13}+x^8+x^7+x^5+x^4+x^2+1$

3. The 24-bit CRC are appended to the data bits in the MSB-first manner.

4. Decoding is identical to encoding except that data fed into the syndrome circuit is 24-bit longer than the information bits at encoding. The dividend is also multiplied by x^{24}. If no error occurs, the remainder should satisfy

$R(x)=x^{22}+x^{21}+x^{19}+x^{18}+x^{16}+x^{15}+x^{11}+x^8+x^5+x^4$ (0x6d8930)

And the parity output word will be 0x9276cf.

In contrast to conventional CRC, this special coding scheme makes the encoder fully identical to the decoder and simplifies the hardware design.

4.8.2 Register Definitions

Register Address	Register Function	Acronym
0X82070000	FCS input data register	FCS_DATA
0X82070004	Input data length indication register	FCS_DLEN
0X82070008	FCS parity output register 1, MSB part	FCS_PAR1
0X8207000C	FCS parity output register 2, LSB part	FCS_PAR2
0X82070010	FCS codec status register	FCS_STAT
0X82070014	FCS codec reset register	FCS_RST

Table 82 ***FCS Registers***

0X82070000 FCS input data register FCS_DATA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DX The data bits input. First write of this register is the starting point of the encode or decode process. X=0…15. The input format is D15$\cdot x^n$+ D14$\cdot x^{n-1}$+ D13$\cdot x^{n-2}$+ … + Dk$\cdot x^k$+ …, thus D15 is the first bit

being pushed into the shift register. If the last data word is less than 16 bits, the rest bits are neglected.

0X82070004 Input data length indication register FCS_DLEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LEN**															
Type	WO															

LEN The MCU specifies the total data length in bits to be encoded or decoded. The data length. A number of multiple-of-8 is required (Number_of_Bytes x 8).

0X82070008 FCS parity output register 1, MSB part FCS_PAR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**P15**	**P14**	**P13**	**P12**	**P11**	**P10**	**P9**	**P8**	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
Type	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0X8207000C FCS parity output register 2, LSB part FCS_PAR2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**P23**	**P22**	**P21**	**P20**	**P19**	**P18**	**P17**	**P16**
Type									RC	RC	RC	RC	RC	RC	RC	RC
Reset									0	0	0	0	0	0	0	0

PX Parity bits output. For FCS_PAR2, bit 8 to bit15 will be filled by zeros when reading. X=0…23. The output format is P23·D^{23}+ P22·D^{22}+ P21·D^{21}+ … + Pk·D^{k}+ …+P1·D^{1}+P0, thus P23 is the earliest bit being popped out from the shift register and first appended to the information bits. In other words, {FCS_PAR2[7:0], FCS_PAR1[15:8], FCS_PAR1[7:0] } is the order of appending parity to data.

0X82070010 FCS codec status register FCS_STAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**BUSY**	**FER**	**RDY**
Type														RC	RC	RC
Reset														0	1	0

BUSY Since the codec works in serial manner and the data word is input in parallel manner, BUSY = 1 indicates that current data word is being processed and write to FCS_DATA is invalid. BUSY = 0 allows write of FCS_DATA during encode or decode process.

0 IDLE

1 BUSY

FER Frame error indication, only for decode mode. FER = 0 means no error occurs and FER = 1 means the parity check has failed. Write of FCS_RST.RST or first write of FCS_DATA will reset this bit to 0.

0 Parity Check Pass

1 Parity Check Fail

RDY When RDY = 1, the encode or decode process has been finished. For encode, the parity data in FCS_PAR1 and FCS_PAR2 are correctly available. For decode, FCS_STAT.FER indication is valid. Write of FCS_RST.RST or first write of FCS_DATA will reset this bit to 0.

0 Process is on-going.

 © 2009 MediaTek Inc.

1 Process is finished.

0X82070014 FCS codec reset register FCS_RST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													EN_DE	PAR	BIT	RST
Type													WO	WO	WO	WO

RST RST = 0 resets the CRC coprocessor. Before setup of FCS codec, the MCU needs to set RST = 0 to flush the shift register content before encode or decode.

0 Reset

1 Reserved

BIT BIT = 0 means not to invert the bit order in a byte of data words when the codec is running. BIT = 1 means the bit order in a byte written in FCS_DATA should be reversed.

0 Not invert the bit order of data words

1 Invert the bit order of data

PAR PAR = 0 means not to invert the bit order in a byte of parity words when the codec is running, include reading of FCS_PAR1 and FCS_PAR2. PAR = 1 means bit order of parity words should be reversed, in decoding or encoding.

0 Not invert the bit order of data words

1 Invert the bit order of data

EN_DE EN_DE = 0 means encode; EN_DE = 1 means decode

0 Encode

1 Decode

4.9 GPRS Cipher Unit

4.9.1 General Description

The unit implements the GPRS encryption/decryption scheme that accelerates the computation of encryption and decryption GPRS pattern. The block accelerates the computation of the key stream. However the bit-wise encryption/decryption of the data is still done by the MCU.

Both GEA, GEA2 and GEA3 are supported.

Register Address	Register Function	Acronym
0x82080000	GPRS Encryption Algorithm Control Register	GCU_CON
0x82080004	GPRS Encryption Algorithm Status Register	GCU_SAT
0x82080008	GPRS Secret Key Kc 0 Register	GCU_SKEY0
0x8208000C	GPRS Secret Key Kc 1 Register	GCU_SKEY1
0x82080010	GPRS Message Key Register	GCU_MKEY
0x82080014	GPRS Ciphered Data Register	GCU_CDATA

Table 83 ***GCU Registers***

4.9.2 Register Definitions

0x82080000 GPRS Encryption Algorithm Control Register GCU_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name										**RBO**	**KS**		**SINIT**	**DIR**	**ALG**	
Type										R/W	R/W		WO	R/W	R/W	
Reset										0	10		0	0	0	

This register controls the key generation function of the GPRS Encryption Algorithm.

ALG Choose the encryption/decryption algorithm.

00 = GEA

01 = GEA2

10 = GEA3

11 = Reserved

DIR The DIRECTION input of the GPRS Encryption Algorithm.

SINIT Start initialization. The MCU writes 1 to start initialization. The bit is always read at 0.

KS Control the read access. 00 = byte access, 01 = half word (16 bits) access, 10 = word access, 11 reserved. Default value is 10.

RBO Reversal Byte Order bit. If the bit was set to 1, the byte order of GCU_SKEY0, GCU_SKEY1, GCU_MKEY in write operation and GCU_SKEY0, GCU_SKEY1, GCU_MKEY, GCU_CKEY in read operation would be the reverse of baseband processor, and if the bit was 0, the behavior would be the same as baseband processor. Byte-order of GCU_CON and GCU_SAT is not affected. The default value is 0 which is different from that in MT6218B.

0x82080004 GPRS Encryption Algorithm Status Register GCU_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**STAT**													**KEY_COM**	**INIT**	
Type	RO													RO	RO	
Reset	110													0	0	

This register shows the status of the GPRS Encryption unit.

INIT Initialization flag.

0 Otherwise

1 GCU is currently performing the initialization phase of GEA or GEA2.

KEY_COM Key-stream computation.

0 a new key is available or the GCU is in initialization phase of GEA , GEA2 or GEA3

1 the GCU is computing new key stream for GEA, GEA2, or GEA3

STAT The state of GCU core of GEA and GEA2. For debug purpose

.

0x82080008 GPRS Secret Key Kc 0 Register GCU_SKEY0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**KC[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KC[15:0]**															
Type	R/W															
Reset	0															

0x8208000C GPRS Secret Key Kc 1 Register GCU_SKEY1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**KC[63:48]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**KC[47:32]**															
Type	R/W															
Reset	0															

KC This set of registers shall be programmed with the GPRS Encryption Algorithm secret key.

0x82080010 GPRS Message Key Register GCU_MKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MKEY[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MKEY[15:0]**															
Type	R/W															
Reset	0															

MKEY This register shall be programmed with the “message key” for the GPRS Encryption Algorithm.

0x82080014 GPRS Ciphered DATA Register GCU_CDATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CDATA[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CDATA[15:0]**															
Type	RO															

CDATA The register contains the key stream. GCU will continue to generate next word of key while current word of key is removed.

 © 2009 MediaTek Inc.

4.9.3 Programming Guide

To trigger the hardware, all register fields in GCU_SKEY0, GCU_SKEY1, and GCU_MKEY must be well informed. Then program GCU_STA to kick-off hardware operation.. Then confirm the KEY_COM register to be high before read-back the CDATA.

4.10 MD2GSYS CONFIG Register

In addition to the Pause Mode capability while in the Standby State, the software program can also put each peripheral independently into Power Down Mode while in the Active State by gating off their clock. The typical logic implementation is depicted in **Figure 13**. For all configuration bits, 1 signifies that the function is in Power Down Mode, and 0 means the function is in the Active Mode.

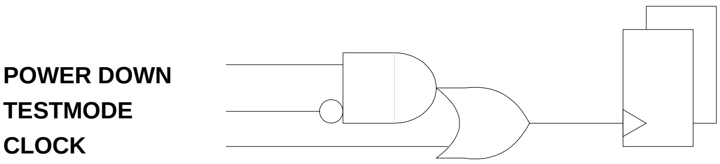


Figure 91 ***Power Down Control at Block Level***

REGISTER ADDRESS	REGISTER NAME	SYNONYM
8200_0000h	Clock Gating Control Register 0	MD2GSYS_CG_CON0
8200_0004h	Clock Gating Control Register 1	MD2GSYS_CG_CON1
8200_0010h	Clock Gating Set Register 0	MD2GSYS_CG_SET0
8200_0014h	Clock Gating Set Register 1	MD2GSYS_CG_SET1
8200_0020h	Clock Gating Clear Register 0	MD2GSYS_CG_CLR0
8200_0024h	Clock Gating Clear Register 1	MD2GSYS_CG_CLR1
8200_0040h	DSP Clock DCM Control Register	DSPCLK_CON
8200_0100h	Memory Delsel Control Regsiter 0 (Used by Hardware)	MD2GSYS_DELSEL0
8200_0104h	Memory Delsel Control Regsiter 1 (Used by Hardware)	MD2GSYS_DELSEL1
8200_0108h	Memory Delsel Control Regsiter 2 (Used by Hardware)	MD2GSYS_DELSEL2
8200_010ch	Memory Delsel Control Regsiter 3 (Used by Hardware)	MD2GSYS_DELSEL3
8200_0110h	Memory Delsel Control Regsiter 4 (Used by Hardware)	MD2GSYS_DELSEL4

 © 2009 MediaTek Inc.

Table 84 ***APB Bridge Register Map***

4.10.1 Register Definitions

8200_0000h Clock Gating Control Status Register MD2GSYS_CG_CON0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**IRDBG2**
Type																RO
Reset																1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRDBG1**			**DIV**	**GCC**	**BFE**	**VFE**		**FCS**	**APC**		**BPI**	**BSI**		**IRDMA**	**GCU**
Type	RO			RO	RO	RO	RO		RO	RO		RO	RO		RO	RO
Reset	1			1	1	1	1		1	1		1	1		1	1

MD2G sub-system power down control status register (read only), value 1 represents power down.

GCU Status of the GCU Controller Power Down.

IRDMA Status of the IRDMA Power Down.

RTC Status of the RTC Power Down.

BSI Status of the BSI Power Down. This status is not taken effect until both tdma_evtval and qbit_en are asserted.

BPI Status of the BPI Power Down. This status is not taken effect until both tdma_evtval and qbit_en are asserted.

APC Status of the APC Power Down. This status is not taken effect until both tdma_evtval and qbit_en are asserted.

FCS Status of the FCS Power Down.

VFE Status of the Audio Front End of VBI Power Down.

BFE Status of the Base-Band Front End Power Down.

GCC Status of the GCC Power Down.

DIV Status of the Divider Power Down.

IRDBG1 Status of the IRDBG1 Power Down.

IRDBG2 Status of the IRDBG2 Power Down.

8200_0004h Clock Gating Control Status Register for radio blocks MD2GSYS_CG_CON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**IRDBG2**
Type																RO
Reset																1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRDBG1**			**DIV**	**GCC**	**BFE**	**VFE**		**FCS**	**APC**		**BPI**	**BSI**		**IRDMA**	**GCU**
Type	RO			RO	RO	RO	RO		RO	RO		RO	RO		RO	RO
Reset	1			1	1	1	1		1	1		1	1		1	1

MD2G sub-system radio block power down direct status, value 1 represents power down.

© 2009 MediaTek Inc.

BSI Status of the BSI Power Down. This status takes effect immediately.
BPI Status of the BPI Power Down. This status takes effect immediately.
APC Controls the APC Power Down. This status takes effect immediately.
OTHERS Same as MD2GSYS_CG_CON0.

8200_0010h Clock Gating Set Register MD2GSYS_CG_SET0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																IRDBG2
Type																WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRDBG1			DIV	GCC	BFE	VFE		FCS	APC		BPI	BSI		IRDMA	GCU
Type	WO			WO	WO	WO	WO		WO	WO		WO	WO		WO	WO

MD2G sub-system power down set register, value 1 represents power down. For all registers addresses listed above, writing to the corresponding "SET" register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.
Eg.
If CG_CON0 = 16'h0F0F,
Writing CG_SET0 = 16'F0F0 will result in CG_CON0 = 16'hFFFF.

GCU Set the GCU Controller Power Down.
IRDMA Set the IRDMA Power Down.
RTC Set the RTC Power Down.
BSI Set the BSI Power Down. This set value is not taken effect until both tdma_evtval and qbit_en are asserted.
BPI Set the BPI Power Down. This set value is not taken effect until both tdma_evtval and qbit_en are asserted.
APC Set the APC Power Down. This set value is not taken effect until both tdma_evtval and qbit_en are asserted.
FCS Set the FCS Power Down.
VFE Set the Audio Front End of VBI Power Down.
BFE Set the Base-Band Front End Power Down.
GCC Set the GCC Power Down.
DIV Set the Divider Power Down.
IRDBG1 Set the IRDBG1 Power Down.
IRDBG2 Set the IRDBG2 Power Down.

8200_0014h Clock Gating Set Register for radio blocks MD2GSYS _CG _SET1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										APC		BPI	BSI			

Type										WO		WO	WO			

MD2G sub-system power down direct set register, value 1 represents power down.

BSI Set the BSI Power Down. This setting takes effect immediately.

BPI Set the BPI Power Down. This setting takes effect immediately.

APC Set the APC Power Down. This setting takes effect immediately.

8200_0020h Clock Gating Clear Register MD2GSYS_CG_CLR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**IRDBG2**
Type																WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRDBG1**			**DIV**	**GCC**	**BFE**	**VFE**		**FCS**	**APC**		**BPI**	**BSI**		**IRDMA**	**GCU**
Type	WO			WO	WO	WO	WO		WO	WO		WO	WO		WO	WO

MD2G sub-system power down set register, value 1 represents power down. For all registers addresses listed above, writing to the corresponding "Clear" register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.

Eg.

If CG_CON0 = 16'hFFFF,

Writing CG_CLR0 = 16'F0F0 will result in CG_CON0 = 16'h0F0F.

GCU Clear the GCU Controller Power Down.

IRDMA Clear the IRDMA Power Down.

RTC Clear the RTC Power Down.

BSI Clear the BSI Power Down. This clear value is not taken effect until both tdma_evtval and qbit_en are asserted.

BPI Clear the BPI Power Down. This clear value is not taken effect until both tdma_evtval and qbit_en are asserted.

APC Clear the APC Power Down. This clear value is not taken effect until both tdma_evtval and qbit_en are asserted.

FCS Clear the FCS Power Down.

VFE Clear the Audio Front End of VBI Power Down.

BFE Clear the Base-Band Front End Power Down.

GCC Clear the GCC Power Down.

DIV Clear the Divider Power Down.

IRDBG1 Clear the IRDBG1 Power Down.

IRDBG2 Clear the IRDBG2 Power Down.

8200_0024h Clock Gating Clear Register for radio blocks MD2GSYS _CG _CLR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**APC**		**BPI**	**BSI**			
Type										WO		WO	WO			

MD2G sub-system power down direct clear register.

BSI Clear the BSI Power Down. This setting takes effect immediately.

BPI Clear the BPI Power Down. This setting takes effect immediately.

APC Clear the APC Power Down. This setting takes effect immediately.

8200_0040 DSP Clock Frequency Setting DSPCLK_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**DSP2_FSEL**				**DSP1_FSEL**		
Type										R/W				R/W		
Reset										111				111		

The register defines the output frequency of dual DSP DCM (dynamic clock management) for DSP1 and DSP2. Totally 7 levels are provided, ranging from 13MHz to 104MHz (default).

DSP1_FSEL DSP1 clock rate is (DSP1_FSEL + 1) x 13 MHz

DSP2_FSEL DSP2 clock rate is (DSP2_FSEL + 1) x 13 MHz

8200_0100h Memory delsel control register 0 MD2GSYS_DELSEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SC60**				**SP00**				**SP10**				**SP11**			
Type	R/W				R/W				R/W				R/W			
Reset	1011				1011				1011				1011			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SP40**				**SP41**				**SP50**				**SP51**			
Type	R/W				R/W				R/W				R/W			
Reset	1011				1011				1011				1011			

8200_0104h Memory delsel control register 1 MD2GSYS_DELSEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SC10**				**SC11**				**SC20**				**SC21**			
Type	R/W				R/W				R/W				R/W			
Reset	1011				1011				1011				1011			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SC30**				**SC31**				**SC50**				**SC51**			
Type	R/W				R/W				R/W				R/W			
Reset	1011				1011				1011				1011			

8200_0108h Memory delsel control register 2 MD2GSYS_DELSEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SD21**		**SD30**		**SD31**		**SP30**		**SP31**		**SPN3**		**MCN0**			
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W			
Reset	11		11		11		11		10		10		1011			

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MCN1**				**MPN0**				**SC00**				**SC01**			
Type	R/W				R/W				R/W				R/W			
Reset	1011				1011				1011				1011			

8200_010ch Memory delsel control register 3 MD2GSYS_DELSEL3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SHRAMD2_2**		**VFE**		**MCN2**		**MD00**		**MD01**		**MD02**		**MD03**		**MD04**	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	10		10		10		10		10		10		10		11	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MPN1**		**SC40**		**SC41**		**SD01**		**SD10**		**SD11**		**SD12**		**SD20**	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset	11		11		11		10		10		11		10		11	

8200_0110h Memory delsel control register 4 MD2GSYS_DELSEL4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**BSI**		**VITERBI**		**EQLZR**		**PREFILTER**		**SHRAMD1**		**SHRAMD2_1**	
Type	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
Reset					10		11		01		01		10		10	

4.11 Timing Generator

Timing is the most critical issue in GSM/GPRS applications. The TDMA timer provides a simple interface for the MCU to program all the timing-related events for receive event control, transmit event control and the timing adjustment. Detailed descriptions are mentioned in Section 4.11.1.

4.11.1 TDMA timer

The TDMA timer unit is composed of three major blocks: Quarter bit counter, Signal generator and Event registers.

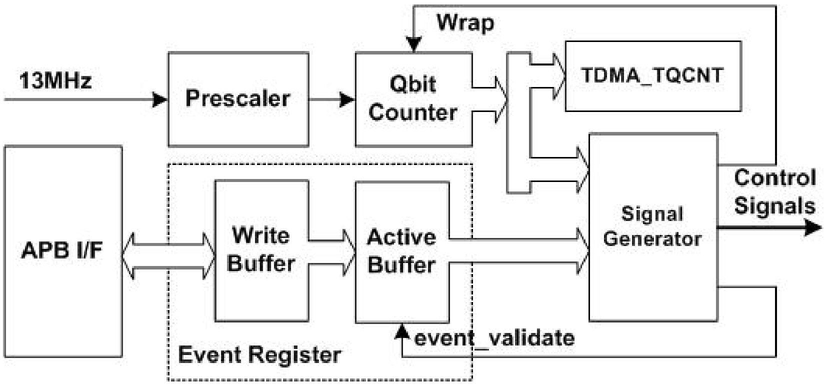
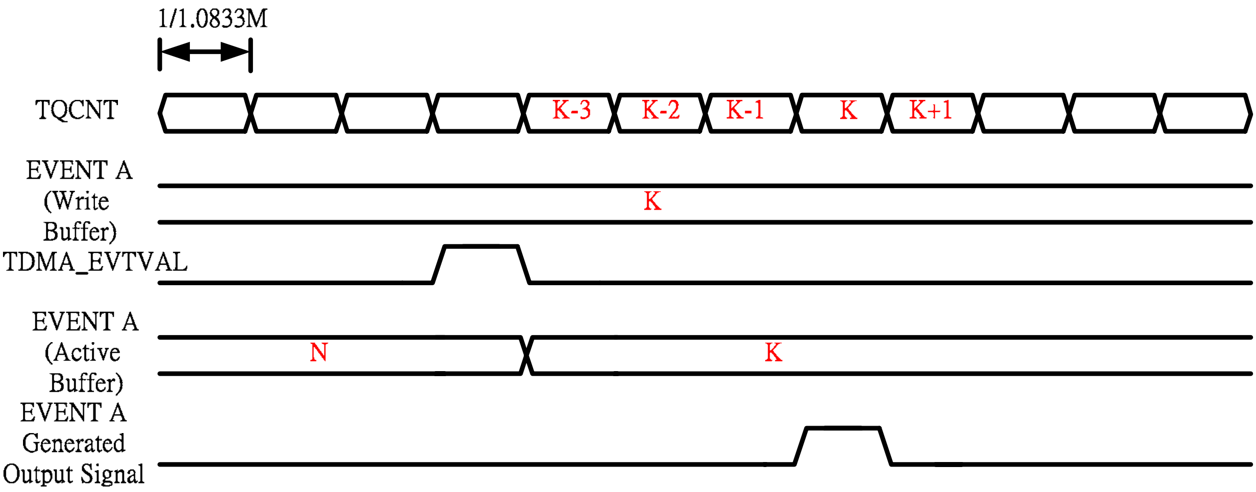


Figure 92 ***The block diagram of TDMA timer***

By default, the quarter-bit counter continuously counts from 0 to the wrap position. In order to apply to cell synchronization and neighboring cell monitoring, the wrap position can be changed by the MCU to shorten or lengthen a TDMA frame. The wrap position is held in the TDMA_WRAP register and the current value of the TDMA quarter bit counter may be read by the MCU via the TDMA_TQCNT register.

The signal generator handles the overall comparing and event-generating processes. When a match has occurred between the quarter bit counter and the event register, a predefined control signal is generated. These control signals may be used for on-chip and off-chip purposes. Signals that change state more than once per frame make use of more than one event register.



The event registers are programmed to contain the quarter bit position of the event that is to occur. The event registers are double buffered. The MCU writes into the write buffers of the registers, and the event TDMA_EVTVAL trigger HW to transfer the data from the write buffers to the active buffers. Caution: values in the active buffers are updated at the end of qbit count (TDMA_EVTVAL). The TDMA_EVTVAL signal itself may be programmed at any quarter bit position. These event registers could be classified into four groups:

On-chip Control Events

TDMA_EVTVAL

This event allows the data values written by the MCU to pass through to the active buffers.

 © 2009 MediaTek Inc.

TDMA_WRAP

TDMA quarter bit counter wrap position. This sets the position at which the TDMA quarter bit counter resets back to zero. The default value is 4999, changing this value will advance or retard the timing events in the frame following the next TDMA_EVTVAL signal. Caution: The wrap value of the first frame after the sleep mode will refer to TQWRAP_SM value if SW enables turbo sleep mode.

TDMA_DTIRQ

DSP TDMA interrupt requests. DTIRQ triggers the DSP to read the command from the MCU/DSP Shard RAM to schedule the activities that will be executed in the current frame.

TDMA_CTIRQ1/CTIRQ2

MCU TDMA interrupt requests. CTIRQx triggers the ARM to schedule the activities that will be executed in the next frame.

TDMA_AUXADC [1:0]

This signal triggers the monitoring ADC to measure the voltage, current, temperature, device id etc..

TDMA_AFC [3:0]

This signal powers up the automatic frequency control DAC for a programmed duration after this event.

Note: For both MCU and DSP TDMA interrupt requests, these signals are all active Low during one quarter bit duration and they should be used as edge sensitive events by the respective interrupt controllers.

On-chip Receive Events

TDMA_BDLON [5:0]

These registers are a set of six which contain the quarter bit event that initiates the receive window assertion sequence which powers up and enables the receive ADC, and then enables loading of the receive data into the receive buffer.

TDMA_BDLOFF [5:0]

These registers are a set of six which contain the quarter bit event that initiates the receive window de-assertion sequence which disables loading of the receive data into the receive buffer, and then powers down the receive ADC.

TDMA_RXWIN[5:0]

DSP TDMA interrupt requests. TDMA_RXWIN is usually used to initiate the related RX processing including two modes. In single-shot mode, TDMA_RXWIN is generated when the BRXFS signal is de-asserted. In repetitive mode, TDMA_RXWIN will be generated both regularly with a specific interval after BRXFS signal is asserted and when the BRXFS signal is de-asserted.

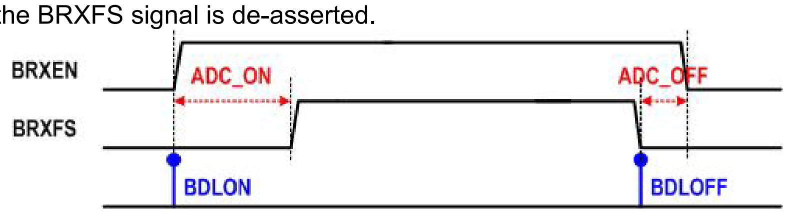


Figure 93 ***The timing diagram of BRXEN and BRXFS***

Note: TDMA_BDLON/OFF event registers, together with TDMA_BDLCON register, generate the corresponding BRXEN and BRXFS window used to power up/down baseband downlink path and control the duration of data transmission to the DSP, respectively.

On-chip Transmit Events

TDMA_APC [6:0]

These registers initiate the loading of the transmit burst shaping values from the transmit burst shaping RAM into the transmit power control DAC.

TDMA_BULON [3:0]

This register contains the quarter bit event that initiates the transmit window assertion sequence which powers up the modulator DAC and then enables reading of bits from the transmit buffer into the GMSK modulator.

TDMA_BULOFF [3:0]

This register contains the quarter bit event that initiates the transmit window de-assertion sequence which disables the reading of bits from the transmit buffer into the GMSK modulator, and then power down the modulator DAC.

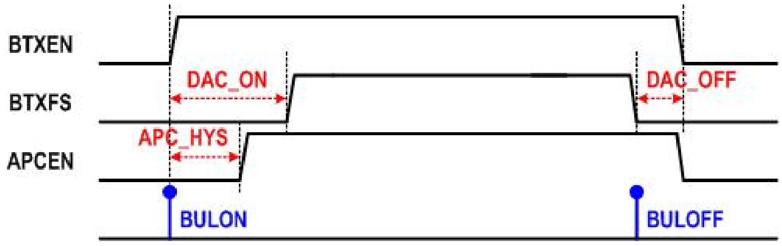


Figure 94 ***The timing diagram of BTXEN and BTXFS***

Note: TDMA_BULON/OFF event registers, together with TDMA_BULCON1, TDMA_BULCON2 register, generate the corresponding BTXEN, BTXFS and APCEN window used to power up/down the baseband uplink path, control the duration of data transmission from the DSP and power up/down the APC DAC, respectively.

Off-chip Control Events

TDMA_BSI [19:0]

The quarter bit positions of these 20 BSI events are used to initiate the transfer of serial words to the transceiver and synthesizer for gain control and frequency adjustment.

TDMA_BPI [41:0]

The quarter bit positions of these 30 BPI events are used to generate changes of state on the output pins to control the external radio components.

4.11.1.1 Register Definitions

0x811f0150 Event Enable Register 0 — TDMA_EVTENA 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AFC3**	**AFC2**	**AFC1**	**AFC0**	**BDL5**	**BDL4**	**BDL3**	**BDL2**	**BDL1**	**BDL0**				**CTIRQ2**	**CTIRQ1**	**DTIRQ**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0				0	0	0

DTIRQ Enable TDMA_DTIRQ

CTIRQ*n* Enable TDMA_CTIRQ*n*

AFC*n* Enable TDMA_AFC*n*

BDL*n* Enable TDMA_BDLON*n* and TDMA_BDLOFF*n*

For all these bits,

0 function is disabled

1 function is enabled

0x811f0154h Event Enable Register 1 — TDMA_EVTENA 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPRS**				**BUL3**	**BUL2**	**BUL1**	**BUL0**		**APC6**	**APC5**	**APC4**	**APC3**	**APC2**	**APC1**	**APC0**

Type	R/W				R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0				0	0	0	0		0	0	0	0	0	0	0

GPRS Indicate which mode is on-going.

- **0** TDMA_APC0 & TDMA_APC1 events are controlled by APC0 & APC1 in the register TDMA_EVTENA1 & TDMA_DTXCON. (GSM mode)
- **1** TDMA_APC0 & TDMA_APC1 events are controlled by APC0 & APC1 in the register TDMA_EVTENA1 only. (GPRS mode)

APC*n* Enable TDMA_APC*n*

BUL*n* Enable TDMA_BULON*n* and TDMA_BULOFF*n*

For all these bits,

- **0** function is disabled
- 1 function is enabled

0x811f0158 Event Enable Register 2 TDMA_EVTENA2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BSI15**	**BSI14**	**BSI13**	**BSI12**	**BSI11**	**BSI10**	**BSI9**	**BSI8**	**BSI7**	**BSI6**	**BSI5**	**BSI4**	**BSI3**	**BSI2**	**BSI1**	**BSI0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x811f015C Event Enable Register 3 TDMA_EVTENA3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**BSI19**	**BSI18**	**BSI17**	**BSI16**
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

BSI*n* BSI event enable control

- **0** Disable TDMA_BSI*n*
- **1** Enable TDMA_BSI*n*

0x811f0160 Event Enable Register 4 TDMA_EVTENA4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BPI15**	**BPI14**	**BPI13**	**BPI12**	**BPI11**	**BPI10**	**BPI9**	**BPI8**	**BPI7**	**BPI6**	**BPI5**	**BPI4**	**BPI3**	**BPI2**	**BPI1**	**BPI0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x811f0164 Event Enable Register 5 TDMA_EVTENA5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BPI31**	**BPI30**	**BPI29**	**BPI28**	**BPI27**	**BPI26**	**BPI25**	**BPI24**	**BPI23**	**BPI22**	**BPI21**	**BPI20**	**BPI19**	**BPI18**	**BPI17**	**BPI16**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BPI*n* BPI event enable control
- **0** Disable TDMA_BPI*n*
- **1** Enable TDMA_BPI*n*

0x811f0168 Event Enable Register 6 TDMA_EVTENA6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							BPI41	BPI40	BPI39	BPI38	BPI37	BPI36	BPI35	BPI34	BPI33	BPI32
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset							0	0	0	0	0	0	0	0	0	0

BPI*n* BPI event enable control
- **0** Disable TDMA_BPI*n*
- **1** Enable TDMA_BPI*n*

0x811f016C Event Enable Register 7 TDMA_EVTENA7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															AUX1	AUX0
Type															R/W	R/W
Reset															0	0

AUX Auxiliary ADC event enable control
- **0** Disable Auxiliary ADC event
- 1 Enable Auxiliary ADC event

0x811f0170 Qbit Timer Offset Control Register TDMA_WRAPOFS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TOI[1:0]	
Type															R/W	
Reset															0	

TOI This register defines the value used to advance the Qbit timer in unit of 1/4 quarter bit; the timing advance will be take place as soon as the TDMA_EVTVAL is occurred, and it will be cleared automatically.

0x811f0174 Qbit Timer Biasing Control Register TDMA_REGBIAS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			TQ_BIAS[13:0]													
Type			R/W													
Reset			0													

TQ_BIAS This register defines the Qbit offset value which will be added to the registers being programmed. It only takes effects on AFC, BDLON/OFF, BULON/OFF, APC, AUXADC, BSI and BPI event registers.

0x811f0180 DTX Control Register TDMA_DTXCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													DTX3	DTX2	DTX1	DTX0
Type													R/W	R/W	R/W	R/W

DTX DTX flag is used to disable the associated transmit signals

0 BULON0~3, BULOFF0~3, APC_EV0 & APC_EV1 are controlled by TDMA_EVTENA1 register

1 BULON0~3, BULOFF0~3, APC_EV0 & APC_EV1 are disabled

0x811f0184 Receive Interrupt Control Register TDMA_RXCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0	RXINTCNT[9:0]									
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W									

RXINTCNT TDMA_RXWIN interrupt generation interval in quarter bit unit

MOD*n* Mode of Receive Interrupts

0 Single shot mode for the corresponding receive window

1 Repetitive mode for the corresponding receive window

0x811f0188 Baseband Downlink Control Register TDMA_BDLCON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADC_ON										ADC_OFF					
Type	R/W										R/W					

ADC_ON BRXEN to BRXFS setup up time in quarter bit unit.

ADC_OFF BRXEN to BRXFS hold up time in quarter bit unit.

0x811f018C Baseband Uplink Control Register 1 TDMA_BULCON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAC_ON								DAC_OFF							
Type	R/W								R/W							

DAC_ON BTXEN to BTXFS setup up time in quarter bit unit.

DAC_OFF BTXEN to BTXFS hold up time in quarter bit unit.

0x811f0190 Baseband Uplink Control Register 2 TDMA_BULCON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									APC_HYS							
Type									R/W							

APC_HYS APCEN to BTXEN hysteresis time in quarter bit unit.

0x811f0194 Frequency Burst Indication Register TDMA_FB_FLAG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											FBDL5	FBDL4	FBDL3	FBDL2	FBDL1	FBDL0
Type											R/W	R/W	R/W	R/W	R/W	R/W

FBDLn Indication of frequency burst for RX window n
The register is double-buffered. The value at the write buffers will be auto-cleared at the next event-validate (TDMA_EVTVAL) and its value will be at the same time loaded to the active buffer. The exact FB indication comes from the active buffer and the corresponding mode in register TDMA_RXCON (Bit15~Bit10). It will be asserted after TDMA_EVTVAL signals if the corresponding FBDLx & TDMA_RXCON[x+10] are set to 1. The FB indication de-assertion only depends TDMA_FB_CLRI and the falling edge of the corresponding RX window.

0x811f0198 Direct Frequency Burst Closing TDMA_FB_CLRI

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																

As long as the register is written, active buffer for FB indication will be reset then therefore the frequency burst indication will be forced to 0.

Address	Type	Width	Reset Value	Name	Description
0x80000000	R	[13:0]	—	TDMA_TQCNT	Read quarter bit counter
0x811f0004	R/W	[13:0]	0x1387	TDMA_WRAP	Latched Qbit counter reset position
0x811f0008	R/W	[13:0]	0x1387	TDMA_WRAPIMD	Direct Qbit counter reset position
0x811f000C	R/W	[13:0]	0x0000	TDMA_EVTVAL	Event latch position
0x811f0010	R/W	[13:0]	—	TDMA_DTIRQ	DSP software control
0x811f0014	R/W	[13:0]	—	TDMA_CTIRQ1	MCU software control 1
0x811f0018	R/W	[13:0]	—	TDMA_CTIRQ2	MCU software control 2
0x811f0020	R/W	[13:0]	—	TDMA_AFC0	The 1st AFC control
0x811f0024	R/W	[13:0]	—	TDMA_AFC1	The 2nd AFC control
0x811f0028	R/W	[13:0]	—	TDMA_AFC2	The 3rd AFC control
0x811f002C	R/W	[13:0]	—	TDMA_AFC3	The 4th AFC control
0x811f0030	R/W	[13:0]	—	TDMA_BDLON0	Data serialization of the 1st RX block
0x811f0034	R/W	[13:0]	—	TDMA_BDLOFF0	
0x811f0038	R/W	[13:0]	—	TDMA_BDLON1	Data serialization of the 2nd RX block
0x811f003C	R/W	[13:0]	—	TDMA_BDLOFF1	
0x811f0040	R/W	[13:0]	—	TDMA_BDLON2	Data serialization of the 3rd RX block
0x811f0044	R/W	[13:0]	—	TDMA_BDLOFF2	
0x811f0048	R/W	[13:0]	—	TDMA_BDLON3	Data serialization of the 4th RX block
0x811f004C	R/W	[13:0]	—	TDMA_BDLOFF3	

 © 2009 MediaTek Inc.

0x811f0050	R/W	[13:0]	—	TDMA_BDLON4	Data serialization of the 5th RX block
0x811f0054	R/W	[13:0]	—	TDMA_BDLOFF4	
0x811f0058	R/W	[13:0]	—	TDMA_BDLON5	Data serialization of the 6th RX block
0x811f005C	R/W	[13:0]	—	TDMA_BDLOFF5	
0x811f0060	R/W	[13:0]	—	TDMA_BULON0	Data serialization of the 1st TX slot
0x811f0064	R/W	[13:0]	—	TDMA_BULOFF0	
0x811f0068	R/W	[13:0]	—	TDMA_BULON1	Data serialization of the 2nd TX slot
0x811f006C	R/W	[13:0]	—	TDMA_BULOFF1	
0x811f0070	R/W	[13:0]	—	TDMA_BULON2	Data serialization of the 3rd TX slot
0x811f0074	R/W	[13:0]	—	TDMA_BULOFF2	
0x811f0078	R/W	[13:0]	—	TDMA_BULON3	Data serialization of the 4th TX slot
0x811f007C	R/W	[13:0]	—	TDMA_BULOFF3	
0x811f0090	R/W	[13:0]	—	TDMA_APC0	The 1st APC control
0x811f0094	R/W	[13:0]	—	TDMA_APC1	The 2nd APC control
0x811f0098	R/W	[13:0]	—	TDMA_APC2	The 3rd APC control
0x811f009C	R/W	[13:0]	—	TDMA_APC3	The 4th APC control
0x811f00A0	R/W	[13:0]	—	TDMA_APC4	The 5th APC control
0x811f00A4	R/W	[13:0]	—	TDMA_APC5	The 6th APC control
0x811f00A8	R/W	[13:0]	—	TDMA_APC6	The 7th APC control
0x811f00B0	R/W	[13:0]	—	TDMA_BSI0	BSI event 0
0x811f00B4	R/W	[13:0]	—	TDMA_BSI1	BSI event 1
0x811f00B8	R/W	[13:0]	—	TDMA_BSI2	BSI event 2
0x811f00BC	R/W	[13:0]	—	TDMA_BSI3	BSI event 3
0x811f00C0	R/W	[13:0]	—	TDMA_BSI4	BSI event 4
0x811f00C4	R/W	[13:0]	—	TDMA_BSI5	BSI event 5
0x811f00C8	R/W	[13:0]	—	TDMA_BSI6	BSI event 6
0x811f00CC	R/W	[13:0]	—	TDMA_BSI7	BSI event 7
0x811f00D0	R/W	[13:0]	—	TDMA_BSI8	BSI event 8
0x811f00D4	R/W	[13:0]	—	TDMA_BSI9	BSI event 9
0x811f00D8	R/W	[13:0]	—	TDMA_BSI10	BSI event 10
0x811f00DC	R/W	[13:0]	—	TDMA_BSI11	BSI event 11
0x811f00E0	R/W	[13:0]	—	TDMA_BSI12	BSI event 12
0x811f00E4	R/W	[13:0]	—	TDMA_BSI13	BSI event 13
0x811f00E8	R/W	[13:0]	—	TDMA_BSI14	BSI event 14
0x811f00EC	R/W	[13:0]	—	TDMA_BSI15	BSI event 15
0x811f00F0	R/W	[13:0]	—	TDMA_BSI16	BSI event 16
0x811f00F4	R/W	[13:0]	—	TDMA_BSI17	BSI event 17
0x811f00F8	R/W	[13:0]	—	TDMA_BSI18	BSI event 18

© 2009 MediaTek Inc.

0x811f00FC	R/W	[13:0]	—	TDMA_BSI19	BSI event 19
0x811f0100	R/W	[13:0]	—	TDMA_BPI0	BPI event 0
0x811f0104	R/W	[13:0]	—	TDMA_BPI1	BPI event 1
0x811f0108	R/W	[13:0]	—	TDMA_BPI2	BPI event 2
0x811f010C	R/W	[13:0]	—	TDMA_BPI3	BPI event 3
0x811f0110	R/W	[13:0]	—	TDMA_BPI4	BPI event 4
0x811f0114	R/W	[13:0]	—	TDMA_BPI5	BPI event 5
0x811f0118	R/W	[13:0]	—	TDMA_BPI6	BPI event 6
0x811f011C	R/W	[13:0]	—	TDMA_BPI7	BPI event 7
0x811f0120	R/W	[13:0]	—	TDMA_BPI8	BPI event 8
0x811f0124	R/W	[13:0]	—	TDMA_BPI9	BPI event 9
0x811f0128	R/W	[13:0]	—	TDMA_BPI10	BPI event 10
0x811f012C	R/W	[13:0]	—	TDMA_BPI11	BPI event 11
0x811f0130	R/W	[13:0]	—	TDMA_BPI12	BPI event 12
0x811f0134	R/W	[13:0]	—	TDMA_BPI13	BPI event 13
0x811f0138	R/W	[13:0]	—	TDMA_BPI14	BPI event 14
0x811f013C	R/W	[13:0]	—	TDMA_BPI15	BPI event 15
0x811f0140	R/W	[13:0]	—	TDMA_BPI16	BPI event 16
0x811f0144	R/W	[13:0]	—	TDMA_BPI17	BPI event 17
0x811f0148	R/W	[13:0]	—	TDMA_BPI18	BPI event 18
0x811f014C	R/W	[13:0]	—	TDMA_BPI19	BPI event 19
0x811f01A0	R/W	[13:0]	—	TDMA_BPI20	BPI event 20
0x811f01A4	R/W	[13:0]	—	TDMA_BPI21	BPI event 21
0x811f01A8	R/W	[13:0]	—	TDMA_BPI22	BPI event 22
0x811f01AC	R/W	[13:0]	—	TDMA_BPI23	BPI event 23
0x811f01B0	R/W	[13:0]	—	TDMA_BPI24	BPI event 24
0x811f01B4	R/W	[13:0]	—	TDMA_BPI25	BPI event 25
0x811f01B8	R/W	[13:0]	—	TDMA_BPI26	BPI event 26
0x811f01BC	R/W	[13:0]	—	TDMA_BPI27	BPI event 27
0x811f01C0	R/W	[13:0]	—	TDMA_BPI28	BPI event 28
0x811f01C4	R/W	[13:0]	—	TDMA_BPI29	BPI event 29
0x811f01C8	R/W	[13:0]	—	TDMA_BPI30	BPI event 30
0x811f01CC	R/W	[13:0]	—	TDMA_BPI31	BPI event 31
0x811f01D0	R/W	[13:0]	—	TDMA_BPI32	BPI event 32
0x811f01D4	R/W	[13:0]	—	TDMA_BPI33	BPI event 33
0x811f01D8	R/W	[13:0]	—	TDMA_BPI34	BPI event 34
0x811f01DC	R/W	[13:0]	—	TDMA_BPI35	BPI event 35
0x811f01E0	R/W	[13:0]	—	TDMA_BPI36	BPI event 36

© 2009 MediaTek Inc.

0x811f01E4	R/W	[13:0]	—	TDMA_BPI37	BPI event 37
0x811f01E8	R/W	[13:0]	—	TDMA_BPI38	BPI event 38
0x811f01EC	R/W	[13:0]	—	TDMA_BPI39	BPI event 39
0x811f01F0	R/W	[13:0]	—	TDMA_BPI40	BPI event 40
0x811f01F4	R/W	[13:0]	—	TDMA_BPI41	BPI event 41
0x811f0400	R/W	[13:0]	—	TDMA_AUXEV0	Auxiliary ADC event 0
0x811f0404	R/W	[13:0]	—	TDMA_AUXEV1	Auxiliary ADC event 1
0x811f0150	R/W	[15:0]	0x0000	TDMA_EVTENA0	Event Enable Control 0
0x811f0154	R/W	[15:0]	0x0000	TDMA_EVTENA1	Event Enable Control 1
0x811f0158	R/W	[15:0]	0x0000	TDMA_EVTENA2	Event Enable Control 2
0x811f015C	R/W	[3:0]	0x0000	TDMA_EVTENA3	Event Enable Control 3
0x811f0160	R/W	[15:0]	0x0000	TDMA_EVTENA4	Event Enable Control 4
0x811f0164	R/W	[13:0]	0x0000	TDMA_EVTENA5	Event Enable Control 5
0x811f0168	R/W	[1:0]	0x0000	TDMA_EVTENA6	Event Enable Control 6
0x811f016C	R/W	[11:0]	0x0000	TDMA_EVTENA7	Event Enable Control 7
0x811f0170	R/W	[1:0]	0x0000	TDMA_WRAPOFS	TQ Counter Offset Control Register
0x811f0174	R/W	[13:0]	0x0000	TDMA_REGBIAS	Biasing Control Register
0x811f0180	R/W	[3:0]	—	TDMA_DTXCON	DTX Control Register
0x811f0184	R/W	[15:0]	—	TDMA_RXCON	Receive Interrupt Control Register
0x811f0188	R/W	[15:0]	—	TDMA_BDLCON	Downlink Control Register
0x811f018C	R/W	[15:0]	—	TDMA_BULCON1	Uplink Control Register 1
0x811f0190	R/W	[7:0]	—	TDMA_BULCON2	Uplink Control Register 2
0x811f0194	R/W	[5:0]	—	TDMA_FB_FLAG	FB indicator
0x811f0198	W		—	TDMA_FB_CLRI	Direct clear of FB indicator
0x811f0264	R	[15:0]	—	TDMA_WRAP_CNT	WRAP Counter for SW LA

Table 85 ***TDMA Timer Register Map***

4.11.1.2 Application Note

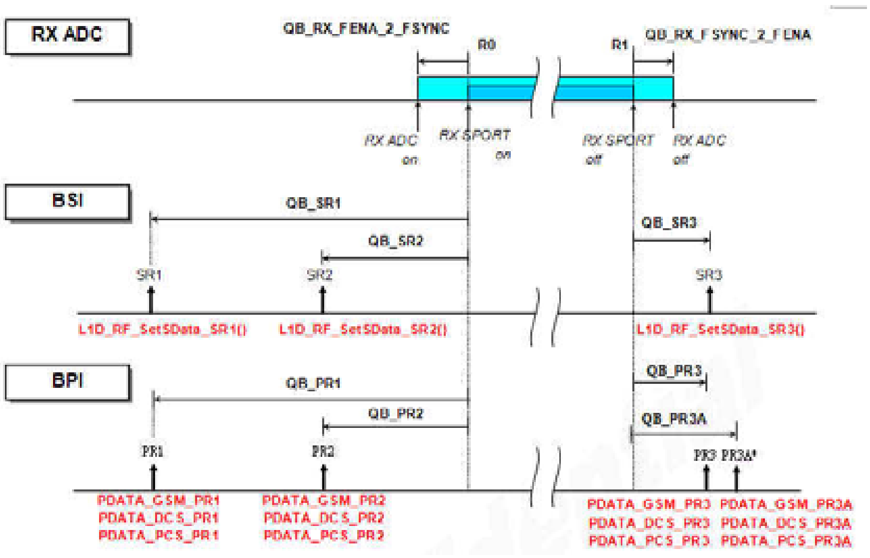


Figure 95 RX Timing Set Example

The TDMA timing and data setting are described in 2 parts. One part is that before turning on RX SPORT to receiving I/Q data. And the other part is after turning off RX SPORT to finish receiving I/Q data. To describe these two parts easily, the timing of turning on RX SPORT is taken as one base named **R0**. And the timing of turning off RX SPORT is taken as one base named **R1**.

RX ADC part:

To setup the timing of RX ADC and SPORT, 2 timings need to be defined in **l1d_custom_rf.h**. The time from RX ADC enabling to RX SPORT turning on (**R0**) is defined as **QB_RX_FENA_2_FSYNC**.The time from RX SPORT turning on (**R1**) to RX ADC disabling is defined as **QB_RX_FSYNC_2_FENA**. The value of this two aliases should be positive or zero. These two values is defined in the register **TDMA_BDLCON**.

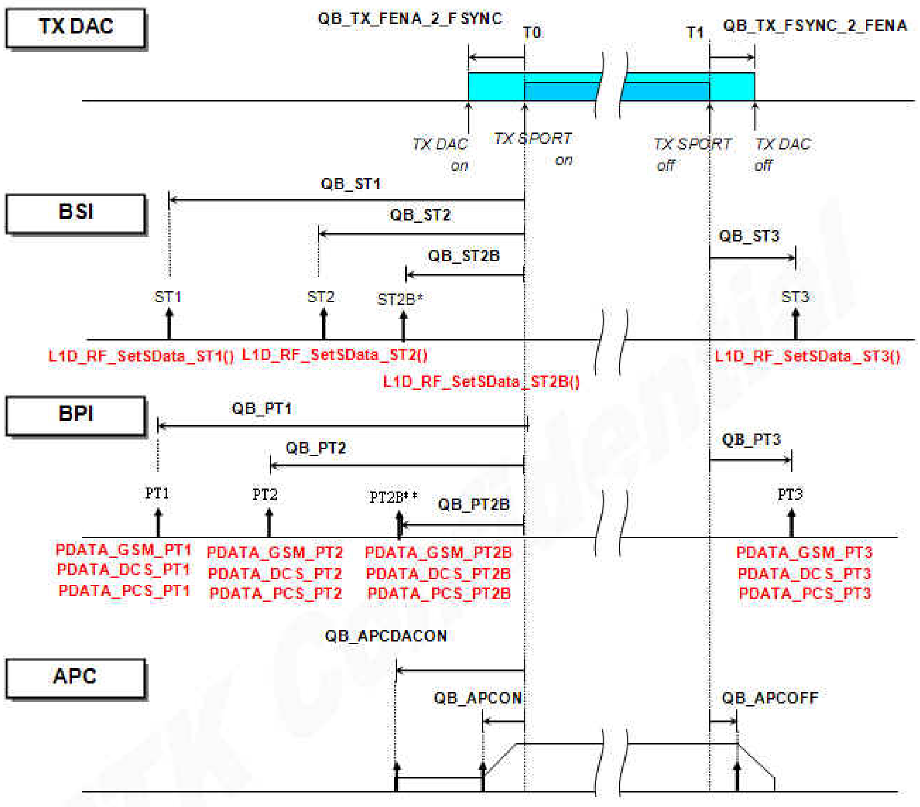
BSI part:

BSI data and events need to be set in serial to a 3-wire base RF module. Each RX window is allocated 3 BSI events. Usually 1'st BSI event is used to warm up the synthesizer and set its N-counter to lock the operation frequency. The 2'nd BSI is used to set the receiving amplifier gain of transceiver. The 3'rd BSI is used to command transceiver entering idle mode. BSI events are defined in the registers **TDMA_BSI0~19**.

BPI part:

The connection of HW signals of BPI data bus and RF module is flexible and depends on customer's design. The setting timing and data setting of BPI bus are used to specify at what time and which BPI states are changed. The BPI data may be varied by the operation band, so the dedicate BPI data of each band should be defined. The states transient of BPI signals are decided by the time of event, therefore the active time and the BPI states for each band shall be defined. BPI events are defined in the registers **TDMA_BPI0~41**.

 © 2009 MediaTek Inc.



TX ADC part:

To setup the timing of TX DAC and SPORT, 2 timings need to be defined in **l1d_custom.h**. The time from TX DAC enabling to TX SPORT turning on (**T0**) is defined as **QB_TX_FENA_2_FSYNC**.The time from TX SPORT turning on (**T1**) to TX DAC disabling is defined as **QB_TX_FSYNC_2_FENA**. The value of this two aliases should be positive or zero. These two values is defined in the register **TDMA_BULCON1**.

BSI part:

BSI data and events need to be set in order to sent serial data to 3-wire devices on RF module. Each TX window is allocated 3 BSI events. Usually 1'st BSI event is used to warm up the set synthesizer and set its N-counter to lock the operation frequency. The 2'nd BSI is used to set the transmit command and indicate the operation band. The 3'rd BSI is used to command transceiver entering idle mode. BSI events are defined in the registers **TDMA_BSI0~19**.

BPI parts:

The setting of BPI bus includes timing and data setting to specify at what time what BPI states are changed. The BPI data may be varied by operation band, so the BPI data of each band should be defined. The 1'st BPI event is usually used to activate the RF components on RF module in transmit mode. The 2'nd BPI event is usually used to select band and switch R/TX. The 3'rd BPI event is usually used to force the RF module into idle mode. BPI events are defined in the registers **TDMA_BPI0~41**.

APC parts:

In addition to TX DAC, TX SPORT, BSI, BPI unit needs to be set, the control of PA is important for TX window. The PA is control by the APC unit of MT62xx. Before the data transmission, APC ramps up the PA to the indicated power level. Data is transmitted at that level. After finishing transmission, APC ramps down the PA.

Before PA ramping up, A DC offset of PA is performed to let PA ramp up smoothly. APC events are defined in the registers **TDMA_APC0~6**.

4.11.2 Slow Clocking Unit

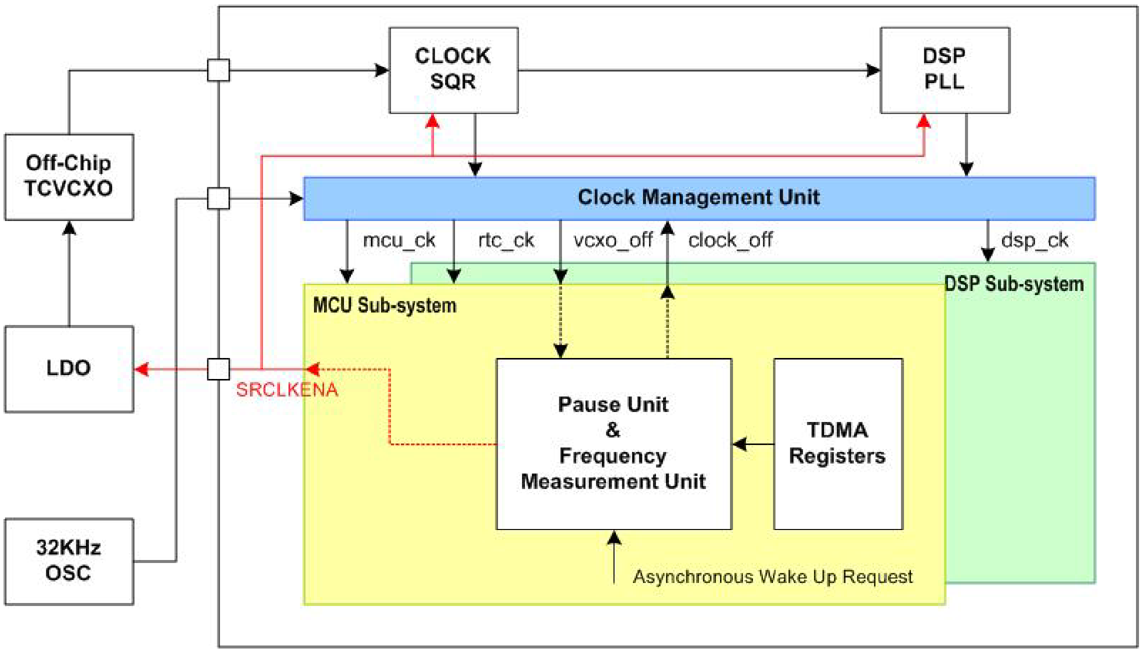


Figure 96 ***The block diagram of the slow clocking unit***

The slow clocking unit is provided to maintain the synchronization to the base-station timing using a 32KHz crystal oscillator while the 13MHz reference clock is switched off. As shown in Figure 39, this unit is composed of frequency measurement unit, pause unit, and clock management unit.

Because of the inaccuracy of the 32KHz oscillator, a frequency measurement unit is provided to calibrate the 32KHz crystal taking the accurate 13MHz source as the reference. The calibration procedure always takes place prior to the pause period.

The pause unit is used to initiate and terminate the pause mode procedure and it also works as a coarse time-base during the pause period.

The clock management unit is used to control the system clock while switching between the normal mode and the pause mode. SRCLKENA is used to turn on/off the clock squarer, DSP PLL and off-chip TCVCXO. CLOCK_OFF signal is used for gating the main MCU and DSP clock, and VCXO_OFF is used as the acknowledgement signal of the CLOCK_OFF request.

4.11.2.1 Register Definitions

0x811f0218 Slow clocking unit control register SM_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**PAUSE_START**	**FM_START**
Type															W	W
Reset															0	0

 © 2009 MediaTek Inc.

FM_START Initiate the frequency measurement procedure

PAUSE_START Initiate the pause mode procedure at the next timer wrap position

0x811f021C Slow clocking unit status register SM_STA

Bit	15	14	13	12	11	10	9	8
Name								PAUSE_ABORT
Type								R
Bit	7	6	5	4	3	2	1	0
Name	SETTLE_CPL	PAUSE_CPL	PAUSE_INT	PAUSE_RQST			FM_CPL	FM_RQST
Type	R	R	R	R			R	R

FM_RQST Frequency measurement procedure is requested

FM_CPL Frequency measurement procedure is completed

PAUSE_RQST Pause mode procedure is requested

PAUSE_INT Asynchronous wake up from pause mode

PAUSE_CPL Pause period is completed

SETTLE_CPL Settling period is completed

PAUSE_ABORT Pause mode is aborted because of the reception of interrupt prior to entering pause mode

0x811f022C Slow clocking unit configuration register SM_CNF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								AP2MD_CCIF	TP	GPT	MSDC	RTC	EINT	KP	SM	FM
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	1	1

FM Enable interrupt generation upon completion of frequency measurement procedure

SM Enable interrupt generation upon completion of pause mode procedure

KP Enable asynchronous wake-up from pause mode by key press

EINT Enable asynchronous wake-up from pause mode by external interrupt

RTC Enable asynchronous wake-up from pause mode by real time clock interrupt

MSDC Enable asynchronous wake-up from pause mode by memory card insertion interrupt

GPT Enable asynchronous wake-up from pause mode by GPT timer

TP Enable asynchronous wake-up from pause mode by touch panel press

AP2MD_CCIF Enable asynchronous wake-up from pause mode by CCIF

0x811f0238 WAKE_PLL_SETTING (TIME & ENABLE) WAKE_PLL_SETTING

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN		TIME													
Type	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1		0	0	0	0	0	0	0	0	1	0	0	0	0	0

TIME The time sleep control generates a reset signal for PLL in the clk setting time.

EN Enable the generation of the reset signal

0 Disable

1 Enable

0x811f023C Jump position of the first frame after sleep mode (TIME & ENABLE) TQINIT_SM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN		TIME													
Type	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	1	0	0	1	1	1	0	0	0	0	1	1	1

TIME The jump position of the first frame after sleep mode. TQCNT will jump from 0 to the pre-defined value, not 1 at the first frame after sleep mode.

EN Enable if the wrap value applies to the first frame after sleep mode.

0 Disable

1 Enable

0x811f0250 Wait-time setting for Power-down strobe TDMA_PDN_SEQUENCE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MD2G_ISO2PWR_T1								MD2G_ISO2RST_T0							
Type	R/W								R/W							
Reset	2								1							

Timing relationship of 2G modem system power on control. Note the SM_CLK_SETTLE should be larger than (MD2G_ISO2PWR_T1 + 1). Otherwise the state machine for the power control gets out of sequence if receiving interrupt during the latency counting.

MD2G_ISO2RST_T0 Latency between isolation assert to reset activated, in unit of 32kHz cycles

MD2G_ISO2PDN_T1 Latency between isolation assert to power-down activated, in unit of 32kHz cycles. MD2G_ISO2PDN_T1 must be larger than MD2G_ISO2RST_T0

0x811f0254 Power down duration TDMA_PDN_DURATION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			MD2G_PDN_DURATION													
Type			R/W													
Reset			0x3ffff													

MD2G_PDN_DURATION Duration of power-down time after MD2G_ISO2PDN_T0 delay, in unit of 32kHz cycles. If the value is too large so that the 26MHz is enabled first (awaken due to PAUSE_CPL, SW_PAUSE_INT or PAUSE_INT), then MD2GSYS automatically powered-on without waiting MD2G_PDN_DURATION expired

0x811f0258 Reset duration TDMA_RST_DURATION

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			MD2G_RST_DURATION													

Type			R/W
Reset			0x3ffff

MD2G_RST_DURATION Duration of reset time after MD2G_ISO2RST_T1 delay, in unit of 32kHz cycles. If the value is too large so that the 26MHz is enabled first (awaken due to PAUSE_CPL, SW_PAUSE_INT or PAUSE_INT), then MD2GSYS automatically reset-released without waiting MD2G_RST_DURATION expired. Typically MD2G_RST_DURATION should be larger than MD2G_PDN_DURATION

0x811f025C Power-down control TDMA_PDN_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MD2G_HW_PDN_EN									MD2G_PDN_SW_EN	MD2G_RST_SW_EN	MD2G_ISO_SW_EN		MD2G_PDN_SEL	MD2G_RST_SEL	MD2G_ISO_SEL
Type	R/W									R/W	R/W	R/W		R/W	R/W	R/W
Reset	0									0	0	0		0	0	0

MD2G_ISO_SEL Select the isolation scheme of MD2GSYS

- **0** Software programming MD2G_ISO_SW_EN
- **1** Hardware control: isolation goes with clock gating signal

MD2G_RST_SEL Select the reset scheme of MD2GSYS

- **0** Software programming MD2G_RST_SW_EN
- **1** Hardware control: reset assert after MD2G_ISO2RST_T0 32kHz cycles, reset de-assert according to either MD2G_RST_DURATION or PAUSE_CPL, or SW_PAUSE_INT/PAUSE_INT

MD2G_PDN_SEL Select the power-down scheme of MD2GSYS

- **0** Software programming MD2G_PDN_SW_EN
- **1** Hardware control: power-down activated after MD2G_ISO2PDN_T1 32kHz cycles, power-on according to either MD2G_RST_DURATION or PAUSE_CPL, or SW_PAUSE_INT/PAUSE_INT

MD2G_ISO_SW_EN Software programming for MD2GSYS isolation timing instance, the isolation signal assert/de-assert right after being programmed without any latency

- **0** Isolation disabled
- **1** Isolation enabled

MD2G_RST_SW_EN Software programming for MD2GSYS reset timing instance, the reset signal assert/de-assert right after being programmed without any latency

- **0** reset de-asserted
- **1** reset asserted

MD2G_PDN_SW_EN Software programming for MD2GSYS power-down timing instance, the power-down signal assert/de-assert right after being programmed without any latency

- **0** power-on
- **1** power-down

MD2G_HW_PDN_EN Enable hardware control sequence for the MD2GSYS isolation, reset and power-down process

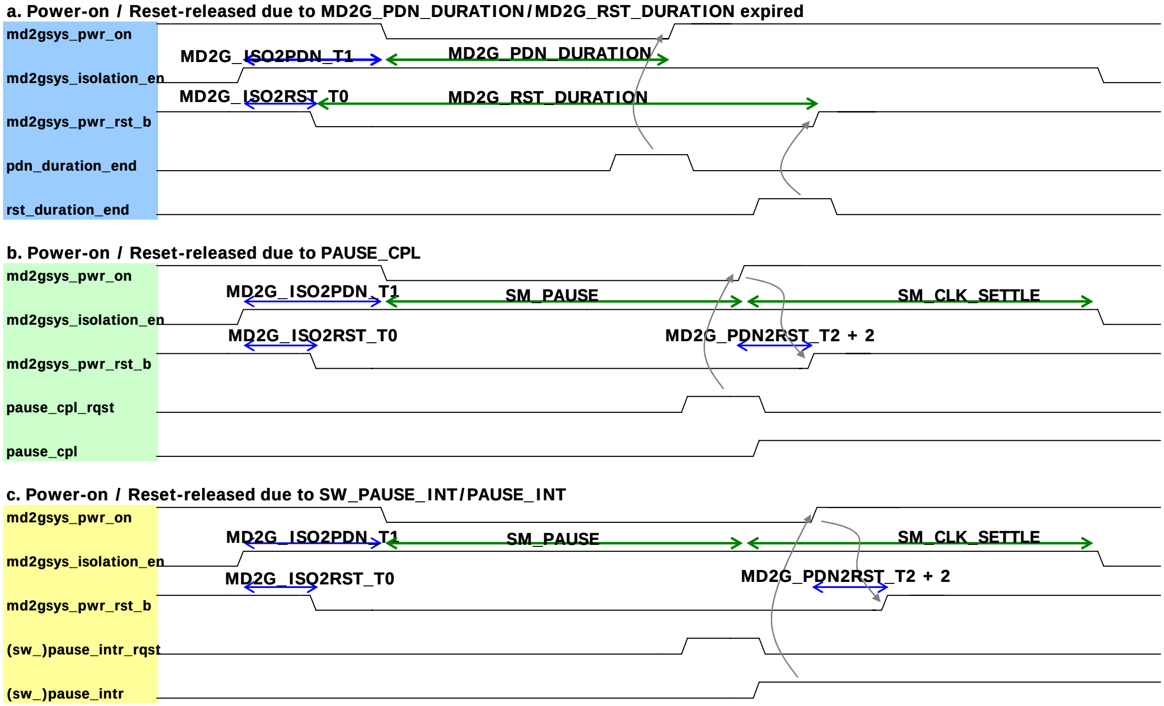
- **0** Hardware control disabled
- **1** Hardware control enabled

0x811f0260 Wait-time setting for RESET strobe TDMA_PDN_SEQUENCE2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MD2G_PDN2RST_T2							
Type									R/W							
Reset									1							

MD2G_PDN2RST_T2 Latency between pwr_on assert to reset released, in unit of 32kHz cycles.
SM_CLK_SETTLE should be larger than (MD2G_PDN2RST_T2 + 2)

Hardware Control Sequence for MD2GSYS power-down:



Address	Type	Width	Reset Value	Name	Description
0x811f0200	R/W	[2:0]	—	SM_PAUSE_M	MSB of pause duration
0x811f0204	R/W	[15:0]	—	SM_PAUSE_L	16 LSB of pause duration
0x811f0208	R/W	[13:0]	—	SM_CLK_SETTLE	Off-chip VCXO settling duration
0x811f020C	R	[2:0]	—	SM_FINAL_PAUSE_M	MSB of final pause count
0x811f0210	R	[15:0]	—	SM_FINAL_PAUSE_L	16 LSB of final pause count
0x811f0214	R	[13:0]	—	SM_QBIT_START	TQ_ COUNT value at the start of the pause
0x811f0218	W	[1:0]	0x0000	SM_CON	SM control register
0x811f021C	R	[7:3,1:0]	0x0000	SM_STA	SM status register
0x811f0220	R/W	[15:0]	—	SM_FM_DURATION	32KHz measurement duration
0x811f0224	R	[9:0]	—	SM_FM_RESULT_M	10 MSB of frequency measurement result
0x811f0228	R	[15:0]	—	SM_FM_RESULT_L	16 LSB of frequency measurement result
0x811f022C	R/W	[4:0]	0x0000	SM_CNF	SM configuration register

0x811f0230	R	[23:0]	0x000000	RTCCOUNT	RTC count
0x811f0238	R/W	[15:0]	0x8020	WAKE_PLL_SETTING	PLL RST time in the clk settling time.
0x811f023C	R/W	[13:0]	0x1387	TQINIT_SM	Jump value of the first frame after sleep mode

4.12 Voice Front-End

4.12.1 General Description

Figure 97 shows the digital circuits block diagram of the audio front-end. The APB register block is an APB peripheral that stores settings from the MCU. The DSP audio port block interfaces with the DSP for control and data communications. The digital filter block performs filter operations for voice band and audio band signal processing. The Digital Audio Interface (DAI) block communicates with the System Simulator for FTA or external Bluetooth modules.

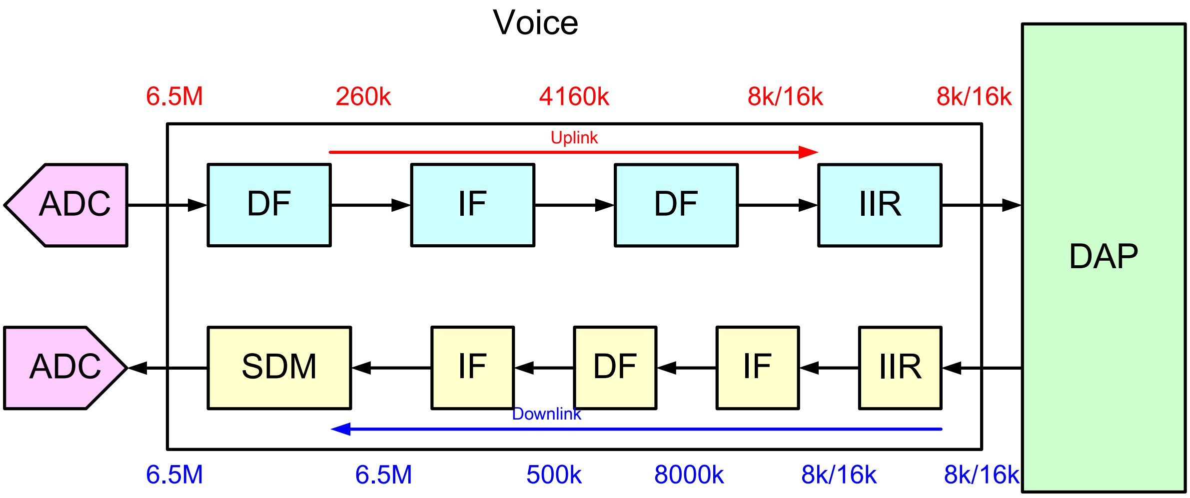


Figure 97 ***Block diagram of digital circuits of the voice front-end***

To communicate with the external Bluetooth module, the master-mode PCM interface and master-mode I2S/EIAJ interface are supported. The clock of PCM interface is 256 kHz while the frame sync is 8 kHz. Both long sync and short sync interfaces are supported. The PCM interface can transmit 16-bit stereo or 32-bit mono 8 kHz sampling rate voice signal. **Figure 98** shows the timing diagram of the PCM interface. Note that the serial data changes when the clock is rising and is latched when the clock is falling.

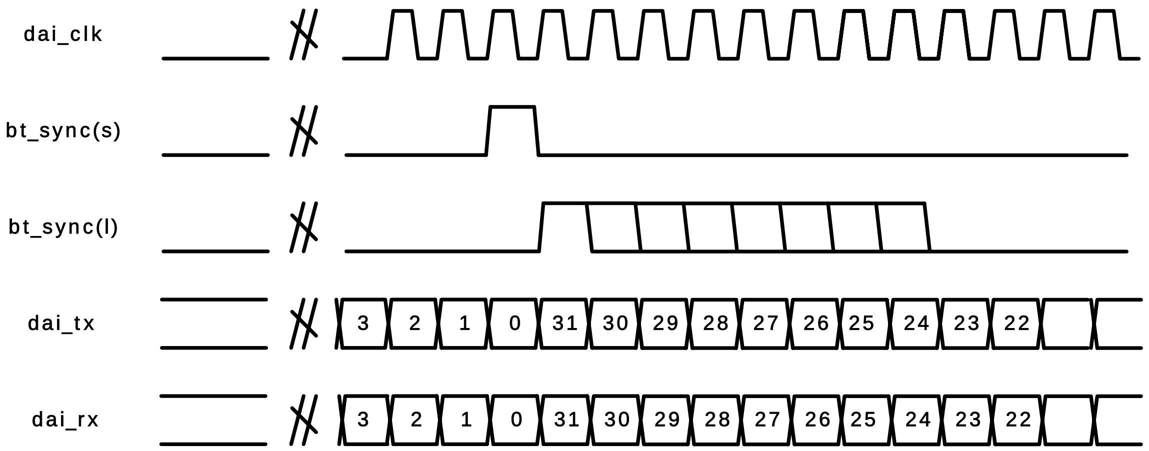


Figure 98 Timing diagram of Bluetooth application

4.12.1.1 DAI, PCM and EDI Pin Sharing

DAI, PCM, and EDI interfaces share the same pins. The pin mapping is listed in **Table 86**.

PIN NAME	DAI	PCM	EDI
DAI_CLK (OUTPUT)	DAI_CLK	PCM_CLK	EDI_CLK
DAI_TX (OUTPUT)	DAI_TX	PCM_OUT	EDI_DAT
DAI_RX (INPUT)	***DAI_RX***	PCM_IN	
BT_SYNC (OUTPUT)	-	PCM_SYNC	EDI_WS

Table 86 ***Pin mapping of DAI, PCM, and EDI interfaces.***

Beside the shared pins, the EDI interface can also use other dedicated pins. With the dedicated pins, PCM and EDI interfaces can operate at the same time.

4.12.2 Register Definitions

MCU APB bus registers in voice front-end are listed as follows. Notes : It can be control either by ARM7 or ARM9. For ARM7, the base address VFE = 820f0000; For ARM9, the base address VFE = 8003B000

Note : All MCU Read/Write of VFE should be done after Voice clock powers on.

0X820F0000 Voice MCU Control Register VFE_VMCU_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																VAFE ON
Type																R/W
Reset																0

MCU sets this register to start AFE voice operation. A synchronous reset signal is issued, then periodical interrupts of 8-KHz frequency are issued. Clearing this register stops the interrupt generation.

VAFEON Turn on audio front-end operations.

0 off

1 on

0X820F000C Voice Analog-Circuit Control Register 1 VFE_VMCU_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VFE_WATER_LV		VIDWA	DSP_CLKRATE	VMODE4K	VAFECLR_EN	VRSDON	VDL_IIRMODE		VUL_IIRMODE		VDLDITH_VAL		VDLDITH_ON
Type			R/W		R/W	R/W	R/W	R/W	R/W	R/W		R/W		R/W		R/W
Reset			00		0	0	0	0	1	00		00		00		0

Set this register current operation mode. Suggested value is 0x0881.

VFE_WATER_LV Voice buffer water level to DMA

VIDWA Voice SDM IDWA enable signal

- **0** close IDWA
- **1** enable IDWA

DSP_CLKRATE DSP clock rate selection

- **0** 104 MHz
- **1** 130 MHz

VMODE4K DSP data mode selection

- **0** 8k mode
- **1** 4k mode

VAFECLR_EN Enable signal to reset voice downlink buffer or not while VAFE is powered down.

- **0** NO reset voice downlink buffer while VAFE is powered down
- **1** Reset voice downlink buffer while VAFE is powered down

VRSDON Turn on the voice-band redundant signed digit function.

- **0** 1-bit 2-level mode
- **1** 2-bit 3-level mode

VDL_IIRMODE Voice downlink IIR coefficients set selection

- **00** 4k : 90Hz, 8k: 180Hz.
- **01** 4k : 160Hz, 8k: 320Hz.
- **10** 4k : 200Hz, 8k: 400Hz
- **11** 4k : 320Hz, 8k: 640Hz

VUL_IIRMODE Voice uplink IIR coefficients set selection

- **00** 4k : 90Hz, 8k: 180Hz.
- **01** 4k : 160Hz, 8k: 320Hz.
- **10** 4k : 200Hz, 8k: 400Hz
- **11** 4k : 320Hz, 8k: 640Hz

VDITHVAL Voice downlink dither scaling setting

- **00** 1/4X
- **01** 1/2X
- **10** 1X
- **11** 2X

VDITHON Turn on the voice downlink dither function.

0 Turn off
1 Turn on

0x820F0010 AFE Voice MCU Control Register 2 AFE_VMCU_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					VTX_CK_PHASE	VRX_CK_PHASE	DWL_OUT_GAIN		DWL_IN_GAIN		VSDM_GAIN					
Type					R/W	R/W	R/W		R/W		R/W					
Reset					0	0	00		00		010000					

Set this register current operation mode. Suggested value is 0x0010.

VTX_CK_PHASE Phase selection of clock to VBITX.
0 Original
1 Inverse

VRX_CK_PHASE Phase selection of clock to VBIRX.
0 Original
1 Inverse

DWL_OUT_GAIN Gain setting at Gain Stage input.
00 1X
01 2X
10 4X
11 1X

DWL_IN_GAIN Gain setting at 8k/16k input.
00 1X
01 1/2X
10 1/4X
11 1X

VSDM_GAIN Gain settings at Voice SDM input.
000000 0/32
000001 1/32
000010 2/32
000011 3/32
•
•
•
100000 32/32
111111 63/32

0X820F0014 Voice DAI Bluetooth Control Register VFE_VDB_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VDAIBT_CLR_EN		VDAION	VBTON	VBTSYNC	VBTSLEN		
Type									R/W		R/W	R/W	R/W	R/W		

Reset									0		0	0	0	000		

Set this register for DAI test mode and Bluetooth application.

DAIBT_CLR_EN Enable signal to reset DAIBT buffer or not while VAFE is powered down.

- **0** NO reset DAIBT buffer while VAFE is powered down
- **1** Reset DAIBT buffer while VAFE is powered down

VDAION Turn on the DAI function.

VBTON Turn on the Bluetooth PCM function.

VBTSYNC Bluetooth PCM frame sync type

- **0**: short
- **1**: long

VBTSLEN Bluetooth PCM long frame sync length = VBTSLEN+1

0X820F0018 Voice Look-Back mode Control Register VFE_VLB_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												VDSPCSMODE	VBYPASSIIR	VDAPINMODE	VINTINMODE	VDECINMODE
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

Set this register for AFE voice digital circuit configuration control. Several loop back modes are implemented for test purposes. Default values correspond to the normal function mode.

VDSPCSMODE DSP COSIM only, to align DATA.

- **0** Normal mode
- **1** DSP COSIM mode

VBYPASSIIR Bypass IIR filter

- **0** Normal mode
- **1** Bypass

VDAPINMODE DSP audio port input mode control

- **0** Normal mode
- **1** Loop back mode

VINTINMODE interpolator input mode control

- **0** Normal mode
- **1** Loop back mode

VDECINMODE decimator input mode control

- **0** Normal mode
- **1** Loop back mode

0X820F0030 Voice DAC SineWave Generator VFE_DAC_TEST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VON		MUTE			AMP_DIV			FREQ_DIV							
Type	R/W		R/W			R/W			R/W							
Reset	0		0			111			0000_0001							

This register is only for analog design verification on audio/voice DACs.

VON Makes voice DAC output the test sine wave.

0 Voice DAC inputs are normal voice samples

1 Voice DAC inputs are sine waves

MUTE Mute switch.

0 Turn on the sine wave output in this test mode.

1 Mute the sine wave output.

AMP_DIV Amplitude setting.

111 full scale

110 1/2 full scale

101 1/4 full scale

100 1/8 full scale

FREQ_DIV Frequency setting, 1 hot.

0000_0001 1X frequency

0000_0010 2X frequency

0000_0100 3X frequency

0000_1000 4X frequency

0001_0000 8X frequency

0010_0000 16X frequency

0100_0000 32X frequency

1000_0000 64X frequency

0X820F0100 VFE AGC Control Register 0 VFE_VAGC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MINPGAGAIN						PGAGAIN						FRFLG	RATKFLG	SATKFLG	AGCON
Type	R/W						R/W						R/W	R/W	R/W	R/W
Reset	001010						101000						0	1	1	1

This register sets the control signals for AGC.

AGCON Switch of the AGC

0 Off

1 On

SATKFLG Sample Attack Flag

0 off

1 on

RATKFLG RMS Attack Flag

0 off

1 on

FRFLG Free Release Flag

0 off

1 on

PGAGAIN PGA gain settings (from -20dB to 43 dB), it is also the maximum PGA gain settings while AGC is on.

000000 -20dB
000001 -19dB
●
●
●
111110 42dB
111111 43dB

MINPGAGAIN minima PGA gain settings (from -20 to 43 dB). PGA gain is always larger than MINPGAGAIN.

000000 -20dB
000001 -19dB
●
●
●
111110 42dB
111111 43dB

0X820F0104 VFE AGC Control Register 1 VFE_VAGC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VAGC_SEL	VAGC_CTRL	ECNTRRLZS				ECNTRRLZF				ECNTRATK			
Type			R/W	R/W	R/W				R/W				R/W			
Reset			1	0	1100				1100				1000			

This register sets the control signals for AGC.

VAGC_SEL Selection of AGC output.

0 bypass AGC.
1 AGC compensation on.

VAGC_CTRL Selection the AGC gain control master.

0 Control by AFE.
1 Control by DSP.

ECNTRATK Attack counter, control attack speed.(unit: N samples@52kHz). Attach will be triggered if N samples amplitude exceed attack threshold (ENTHDATK)

0 always attack, please don’t set to this values.
1~15 N=1~15

ECNTRRLZF Fast release counter, control fast release speed.(unit: N samples@52kHz). Release will be triggered if N samples amplitude lower than slow release threshold (ENTHDRLS)

0 1
1 3
2 7
3 15
4 31
5 63
6 127
7 255

 © 2009 MediaTek Inc.

8 511
9 1023
10 2043
11 4095
12 8191
13 16383
14 32767
15 65535

ECNTRRLZS Slow release counter, control slow release speed.(unit: N samples@52kHz). Release will be triggered if N samples amplitude lower than hysteresis threshold (ENTHDHYS)

0 1
1 3
2 7
3 15
4 31
5 63
6 127
7 255
8 511
9 1023
10 2043
11 4095
12 8191
13 16383
14 32767
15 65535

0X820F0108 VFE AGC Control Register 2 VFE_VAGC_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ERMSFBATTF				ERMSFBATTR				ERMSFBF				ERMSFBR			
Type	R/W				R/W				R/W				R/W			
Reset	1010				0100				1011				0101			

This register sets the control signals for AGC.

ERMSFBR RMS rising factor**.** The larger the number; the slower the signal energy estimation.

0 1x RMS power estimation.
1 2x RMS power estimation.
2 4x RMS power estimation.
3 8x RMS power estimation.

-
-
-

14 16384x RMS power estimation.
15 32768x RMS power estimation.

ERMSFBF RMS falling factor. The larger the number; the slower the signal energy estimation.

- **0** 1x RMS power estimation.
- **1** 2x RMS power estimation.
- **2** 4x RMS power estimation.
- **3** 8x RMS power estimation.
- •
- •
- •
- **14** 16384x RMS power estimation.
- **15** 32768x RMS power estimation.

ERMSFBATTR RMS for Attack rising factor. The larger the number; the slower the signal energy estimation.

- **0** 1x RMS power estimation.
- **1** 2x RMS power estimation.
- **2** 4x RMS power estimation.
- **3** 8x RMS power estimation.
- •
- •
- •
- **14** 16384x RMS power estimation.
- **15** 32768x RMS power estimation.

ERMSFBATTF RMS for Attack falling factor. The larger the number; the slower the signal energy estimation.

- **0** 1x RMS power estimation.
- **1** 2x RMS power estimation.
- **2** 4x RMS power estimation.
- **3** 8x RMS power estimation.
- •
- •
- •
- **14** 16384x RMS power estimation.
- **15** 32768x RMS power estimation.

0X820F010C VFE AGC Control Register 3 VFE_VAGC_CON3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EGAINCOMP_CSS			EGAINCOMP_CSM			EGAINCOMP_CSF			EGAINCOMP_FC_THD			EGAINCOMP_LOWER		EGAINCOMP_UPPER	
Type	R/W			R/W			R/W			R/W			R/W		R/W	
Reset	001			011			011			101			01		01	

This register sets the control signals for AGC.

© 2009 MediaTek Inc.

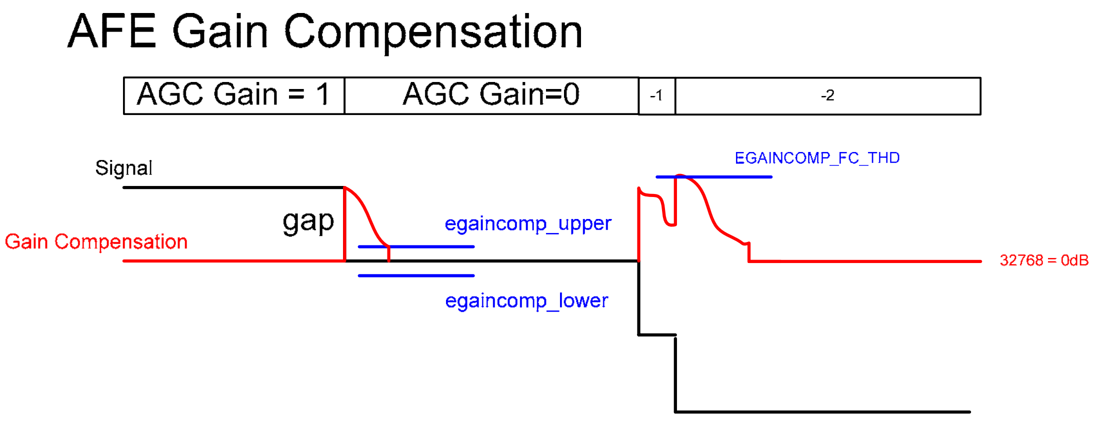


Figure 99 ***Gain Compensation procedures.***

EGAINCOMP_UPPER Gain compensation upper threshold, 32768 = 0dB (**Figure 99** Gain Compensation procedures.)

00 33095
01 33423
10 33751
11 34078

EGAINCOMP_LOWER Gain compensation lower threshold, 32768 = 0dB (**Figure 99** Gain Compensation procedures.)

00 32440
01 32112
10 31784
11 31457

EGAINCOMP_FC_THD Gain compensation convergence threshold (**Figure 99** Gain Compensation procedures.).

000 34406
001 36044
010 37683
011 39321
100 40960
101 42598
110 44236
111 45875

EGAINCOMP_FC_CSF Gain compensation fast converge speed. (While compensation gain is 0.3dB far from 32768, the convergence speed is fast)

000 31948 (8X)
001 31129 (7X)
010 30310 (6X)
011 29491 (5X)
100 28672 (4X)
101 27852 (3X)

 © 2009 MediaTek Inc.

110 27033 (2X)
111 26214 (1X)

EGAINCOMP_FC_CSM Gain compensation converge speed middle. (While compensation gain is 0.15dB ~ 0.3dB from 32768, the convergence speed is middle)

000 32686 (8X)
001 32604 (7X)
010 32552 (6X)
011 32440 (5X)
100 32358 (4X)
101 32276 (3X)
110 32194 (2X)
111 32112 (1X)

EGAINCOMP_FC_CSS Gain compensation converge speed slow. (While compensation gain is inside 0.15dB from 32768, the convergence speed is slow)

000 32751 (8X)
001 32735 (7X)
010 32718 (6X)
011 32702 (5X)
100 32686 (4X)
101 32669 (3X)
110 32653 (2X)
111 32636 (1X)

0X820F0110 VFE AGC Control Register 4 VFE_VAGC_CON4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENTHDATKRMS						ENTHDATK						ESRELWINWIDTH1			
Type	R/W						R/W						R/W			
Reset	000100						000001						1000			

This register sets the control signals for AGC.

ESRELWINWIDTH1 speech release window width for strong VAD

0 10 @ 52kHz samples
1 20 @ 52kHz samples
2 40 @ 52kHz samples
3 80 @ 52kHz samples
4 160 @ 52kHz samples
5 325 @ 52kHz samples
6 650 @ 52kHz samples
7 1300 @ 52kHz samples
8 2600 @ 52kHz samples
9 5200 @ 52kHz samples
10 10000 @ 52kHz samples
11 15000 @ 52kHz samples
12 20000 @ 52kHz samples

 © 2009 MediaTek Inc.

13 25000 @ 52kHz samples
14 30000 @ 52kHz samples
15 32767 @ 52kHz samples

ENTHDATK Attack threshold
[0~63] is map to [-63~0]dB FS

ENTHDATKRMS RMS attack threshold
[0~63] is map to [-63~0]dB FS

0X820F0114 VFE AGC Control Register 5 VFE_VAGC_CON5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENTHDRLS						ENTHDHYS						ESRELWINWIDTH2			
Type	R/W						R/W						R/W			
Reset	001101						000111						1000			

This register sets the control signals for AGC.

ESRELWINWIDTH2 speech release window width for weak VAD
0 10 @ 52kHz samples
1 20 @ 52kHz samples
2 40 @ 52kHz samples
3 80 @ 52kHz samples
4 160 @ 52kHz samples
5 325 @ 52kHz samples
6 650 @ 52kHz samples
7 1300 @ 52kHz samples
8 2600 @ 52kHz samples
9 5200 @ 52kHz samples
10 10000 @ 52kHz samples
11 15000 @ 52kHz samples
12 20000 @ 52kHz samples
13 25000 @ 52kHz samples
14 30000 @ 52kHz samples
15 32767 @ 52kHz samples

ENTHDHYS Hysteresis threshold
[0~63] is map to [-63~0]dB FS

ENTHDRLS Slow release threshold
[0~63] is map to [-63~0]dB FS

0X820F0118 VFE AGC Control Register 6 VFE_VAGC_CON6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EPATTLIMITER				PATTRELD				NATKFLG	PATTRELFLG	ENTHDNOZ					
Type	R/W				R/W				R/W	R/W	R/W					
Reset	0100				0000				1	1	111101					

This register sets the control signals for AGC.

ENTHDNOZ Idle threshold

[0~63] is map to [-63~0]dB FS

PATTRELFLG post attack/release flag

0 off

1 on

NATKFLG noise adaptive attenuation enable attack flag

0 off

1 on

PATTRELD Post attack/release latency

0~15 is map to 0~15 sample @260kHz sampling rate

EPATTLIMITER Post attack limiter

[0,15] is map to [0,-7.5dBFs], the spacing is 0.5dB

0X820F00F0 VFE DMA Read Data Register VFE_DATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VFE_DATA[31:16]															
Type	R/W															
Reset	0000_0000_0000_0000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VFE_DATA[15:0]															
Type	R/W															
Reset	0000_0000_0000_0000															

Virtual FIFO for ARM9 to access voice data from DMA.

VFE_DATA Data received from voice A/D.

4.12.3 DSP Register Definitions

0x640 Voice Uplink Data Register 0 VFE_VUL_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VUL_DAT0															
Type	RO															
Reset	0															

Voice band uplink transmission data register 0. The content of this register is updated by uplink digital filter outputs. This register is read by DSP in an 8K ISR.

0x641 Voice Uplink Data Register 1 VFE_VUL_DAT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VUL_DAT1															
Type	RO															
Reset	0															

Voice band uplink transmission data register 1. The content of this register is updated by uplink digital filter outputs. This register is read by DSP in an 8K ISR if VBYPASSIIR of AFE_LB_CON is set.

0x642 Voice Downlink Data Register 0 VFE_VDL_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDL_DAT0															

Type	WO
Reset	0

Voice band downlink receiving data register 0. This register is written by DSP in an 8K ISR. The content of this register is used as downlink digital filter inputs.

0x643 Voice Downlink Data Register 1 VFE_VDL_DAT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDL_DAT1															
Type	WO															
Reset	0															

Voice band downlink receiving data register 1. This register is written by DSP in an 8K ISR if VBYPASSIIR of AFE_VLB_CON is set. The content of this register is used as downlink digital filter inputs.

0x644 Voice DAI Bluetooth Transmission Data Register 0 VFE_VDBTX_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDBTX_DAT0															
Type	WO															
Reset	0															

DAI Bluetooth transmission data register 0. This register is written by DSP in an 8K ISR if the Bluetooth function is turned on. The content of this register is shifted out to the Bluetooth interface.

0x645 Voice DAI Bluetooth Transmission Data Register 1 VFE_VDBTX_DAT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDBTX_DAT 1															
Type	WO															
Reset	0															

DAI Bluetooth transmission data register 1. This register is written by DSP in an 8K ISR if the corresponding DAI test is set or the Bluetooth function is turned on. The content of this register is shifted out to the SS or Bluetooth interface.

0x646 Voice DAI Bluetooth Receiving Data Register 0 VFE_VDBRX_DAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDBRX_DAT 0															
Type	RO															
Reset	0															

DAI Bluetooth receiving data register 0. This register is read by DSP in an 8K ISR if the Bluetooth function is turned on. The content of this register is shifted in from the Bluetooth interface.

© 2009 MediaTek Inc.

0x647 Voice DAI Bluetooth Receiving Data Register 1 VFE_VDBRX_DAT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDBRX_DAT 1															
Type	RO															
Reset	0															

DAI Bluetooth receiving data register 1. This register is read by DSP in an 8K ISR if the corresponding DAI test is set or the Bluetooth function is turned on. The content of this register is shifted in from the SS or Bluetooth module.

0x648 Voice DAI Bluetooth Control Register VFE_VDSP_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																VDSP_RDY
Type																R/W
Reset																0

DSP sets this register to inform hardware that it is ready for data transmission. In DAI test modes, DSP starts a test by setting vdsp_rdy when speech samples are required or are ready. In normal mode, the DSP asserts this bit to ungate the downlink path data. Otherwise, the downlink data remains zero.

VDSP_RDY Ready indication to start the voice band data path.

- **0** DSP data is not ready.
- **1** DSP data is ready.

0x64A AFE AGC DSP Control AEF_VAGC_VAD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								NADPATT_DBGAIN						NGATEOPEN	VAD2	VAD
Type								R/W						R/W	R/W	R/W
Reset								000000						0	0	0

This register is for DSP to read/write the parameter of AGC.

VAD Strong VAD flag

- **0** off
- **1** on

VAD2 Weak VAD flag

- **0** off
- **1** on

NGATEOPEN noise gate flag

- **0** noise gate close
- **1** noise gate open

NADPATT_DBGAIN noise adaptive attenuation DB gain

[0,63] 0~63dB

0x64B AFE AGC DSP Control AEF_VAGC_CNTR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNTR_REL_FF															TONE FLG
Type	R/W															R/W
Reset	000000000000000															0

This register is for DSP to read/write the parameter of AGC.

TONEFLG Tone flag

0 off

1 on

CNTR_REL_FF Proceed very fast release if N samples value smaller than the fast release threshold

0~32767 N=0~32767 @ 52kHz sampling rate.

0x64C AFE AGC DSP Control1 AEF_VAGC_CNTR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	NCNTRRLZ								NCNTRATK							
Type	RO								RO							
Reset	0000_0000								0000_0000							

This register is for DSP to read the parameter of AGC.

NCNTRRLZ Release counter (in unit of 52kHz/256 sampling rate).

NCNTRATK Attack counter (in unit of 52kHz/16 sampling rate).

0x64D AFE AGC DSP Control AEF_VAGC_STETE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SSTATE		FGAINDB					
Type									RO		RO					
Reset									00		000000					

This register is for DSP to read the parameter of AGC.

FGAINDB Current PGA gain (from 0 to 63 dB).

SSTATE Current AGC state.

0x64E AFE AGC DSP Control AEF_VAGC_RMS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIGRMSATT								SIGRMS							
Type	RO								RO							
Reset	00000000								00000000							

This register is for DSP to read the parameter of AGC.

SIGRMS RMS of signal

SIGRMSATT RMS of signal for attack usage

0x651 VFE AGC DSP GAIN VFE_VAGC_GAIN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											VAGC_GAIN					
Type											R/W					
Reset											000000					

AGC Gain setting by AGC. It is only validate while VAGC_CTRL is set to 1.

4.12.4 Programming Guide

Several cases – including speech call, voice memo record, voice memo playback, melody playback and DAI tests – requires that partial or the whole audio front-end be turned on.

The following are the recommended voice band path programming procedures to turn on voice front-end:

1. MCU programs the VFE_DAI_CON, VFE_LB_CON, VFE_VAG_CON, VFE_VAC_CON0, VFE_VAC_CON1 and VFE_VAPDN_CON registers for specific operation modes. Refer also to the analog chip interface specification.
2. MCU clears the VAFE bit of the PDN_CON2 register to ungate the clock for the voice band path. Refer to the software power down control specification.
3. MCU sets VFE_VMCU_CON to start operation of the voice band path.

The following are the recommended voice band path programming procedures to turn off voice front-end:

1. MCU programs VFE_VAPDN_CON to power down the voice band path analog blocks.
2. MCU clears VFE_VMCU_CON to stop operation of the voice band path.
3. MCU sets VFE bit of PDN_CON2 register to gate the clock for the voice band path.

Note : All MCU Read/Write of VFE should be done after Voice clock powers on.

5 Multimedia Subsystem

MT6516 is a highly integrated multimedia-rich application processor offering an ultra low cost and high performance multimedia solution for Smartphone.

With MT6516, demanding multimedia functions such as recording and playing DVD quality video, video streaming, 5M pixels camera image capturing, 3D gaming, and high resolution true color LCD displaying can be easily implemented without additional multimedia companion chips. Moreover, a high performance video DSP provides extended computing power to support a variety of video codec format and unpredictable multimedia application in the future.

In MT6516, high performance hardware-based multimedia functions enable high quality and real-time Display, Graphics, Image, and Video features for Multimedia-rich Smartphone

Display

- LCD interface support host interface, RGB interface, and MIPI DSI high speed display serial interface
- Support up to 24-bit RGB888 true color panel
- Dual panel (Main and Sub) support
- Display color processing enhancement functions: gamma correction, color correction matrix, dithering
- Image post-processing
- YUV/RGB color space conversion
- Image rotation 90, 180, 270
- TV Out

Graphics

- 3D Graphics, OpenGL ES 1.1 and D3D Mobiles, 3M triangle/sec, 32Mpixel/sec, True color and VGA resolution
- 2D Graphics HW acceleration
- Hardware PNG decoder

Image

- Built-in image signal processing supporting 5M pixel camera sensor
  - Hardwired real-time image processing up to 5M pixels
  - Support YUV/RGB/Bayer image input format
  - 12-bit Parallel interface support
  - 2 channel MIPI CSI-2 high speed camera serial interface support
  - Non-circular based shading compensation
  - Auto defect and defect table compensation
  - Gradation Enhancement
  - Noise reduction
  - Image stabilization
  - Rich image processing function and 32 Special Effect

 © 2009 MediaTek Inc.

- Mechanical shutter support
- Xenon flash support

- Hardware JPEG encode/decode
  - Up to 5M pixels
  - Baseline YUV422/YUV420/YUV411 and grayscale format
  - EXIF/JFIF
- Scaler support arbitrary size Scaling up/down from 1/2048x to 64x with high quality bi-cubic algorithm

Video

- H.264 video decoder D1 30fps
- MPEG4/H.263 video encoder D1 30fps
- MPEG4/H.263 video decoder D1 30fps
- VC-1 video decoder CIF 30fps (by DSP)
- Realvideo decoder CIF 30fps (by DSP)

Audio

- Hardware wavetable coprocessor support at least 128 polyphony stereo real-time synthesis for MIDI
- Sample rate conversion up to 48KHz
- Programmable low-pass IIR filter that can eliminate noise from waveform re-sampling
- Multiple data decompression methods to reduce wavetable size
- Proprietary “Fractional Looping” technology
- I2S interface
- Audio encode: AMR-NB
- Audio decode: MP3, AMR, WB-AMR+, WMA, WMA+, BSAC, AAC, AAC+,

MT6516 Multimedia architecture is shown in Figure 1-1. It contains multimedia accelerators to enhance graphics, display, video, and camera features. High performance graphics bus connecting to 32-bit DDR memory provides very high bandwidth throughput. Hardware multimedia data paths and Image DMA as shown in Figure 1-2 are designed to make data transfer more efficient.

© 2009 MediaTek Inc.

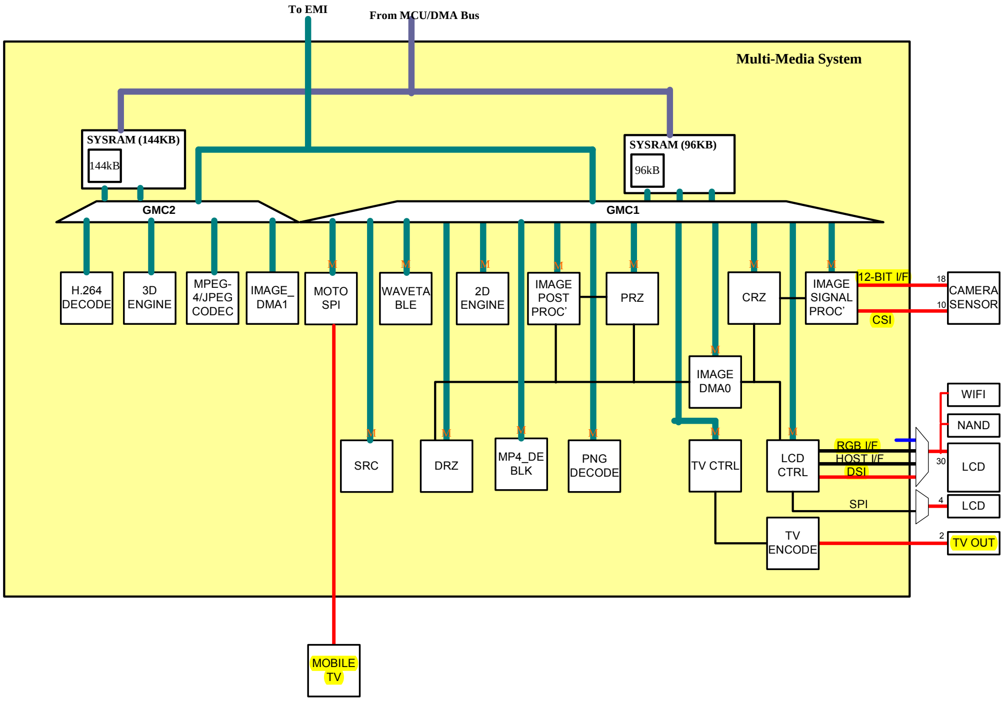


Fig 1-1 MT6516 multimedia architecture

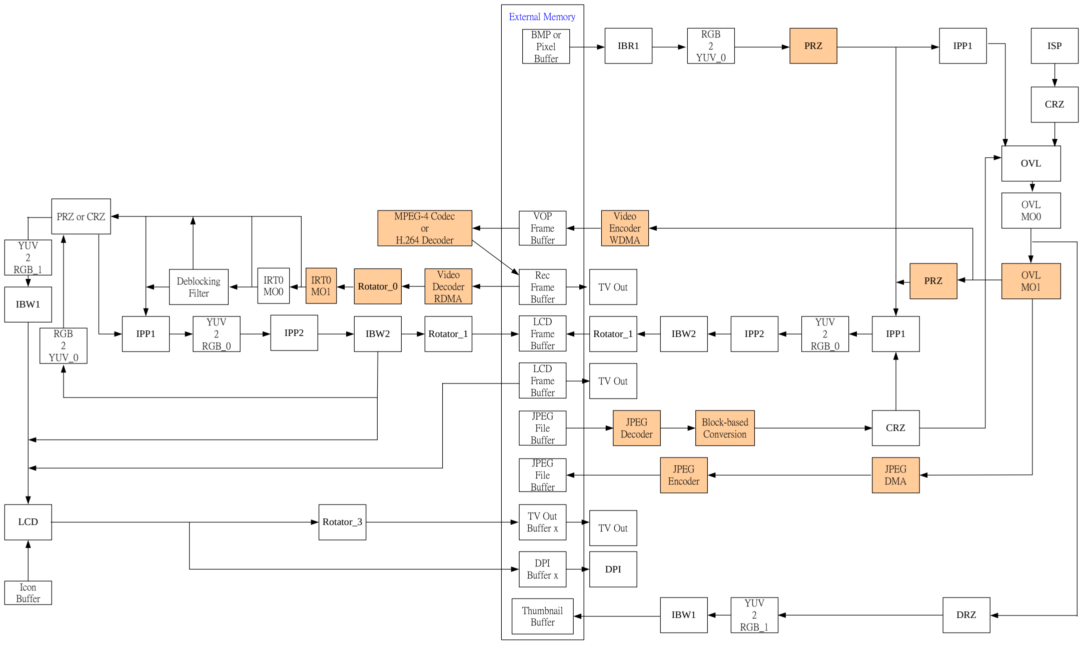


Fig 1-2 MT6516 multimedia datapath

5.1 2D acceleration

5.1.1 2D Engine

5.1.1.1 General Description

To enhance MMI display and gaming experiences, a 2D acceleration engine is implemented. It supports ARGB8888, RGB888, ARGB4444, RGB565 and 8-bpp color modes. Main features are listed as follows:

- Rectangle fill with color gradient.
- Bitblt: multi-Bitblt without transform, 7 rotate, mirror (transparent) Bitblt
- Alpha blending
- ROP3, ROP4
- Font caching: normal font, italic font
- Polygon fill with single color or image pattern
- 1/2/4/8 bit index color Bitblt
- Scaled Bitblt
- Specific output color replacement

MCU can program 2D engine registers via APB. However, MCU has to make sure that the 2D engine is not BUSY before any write to 2D engine registers occurs. An interrupt scheme is also provided for more flexibility. A command parser is implemented for further offloading of MCU. The command queue can be randomly assigned in the system memory, with a maximum depth of 2047 commands. If the command queue is enabled, MCU has to check if the command queue has free space before writing to the command queue. Command queue parser will consume command queue entries upon 2D engine requests. **Figure 100** shows the command queue and 2D engine block diagram. Please refer to the graphic command queue functional specification for more details.

 © 2009 MediaTek Inc.

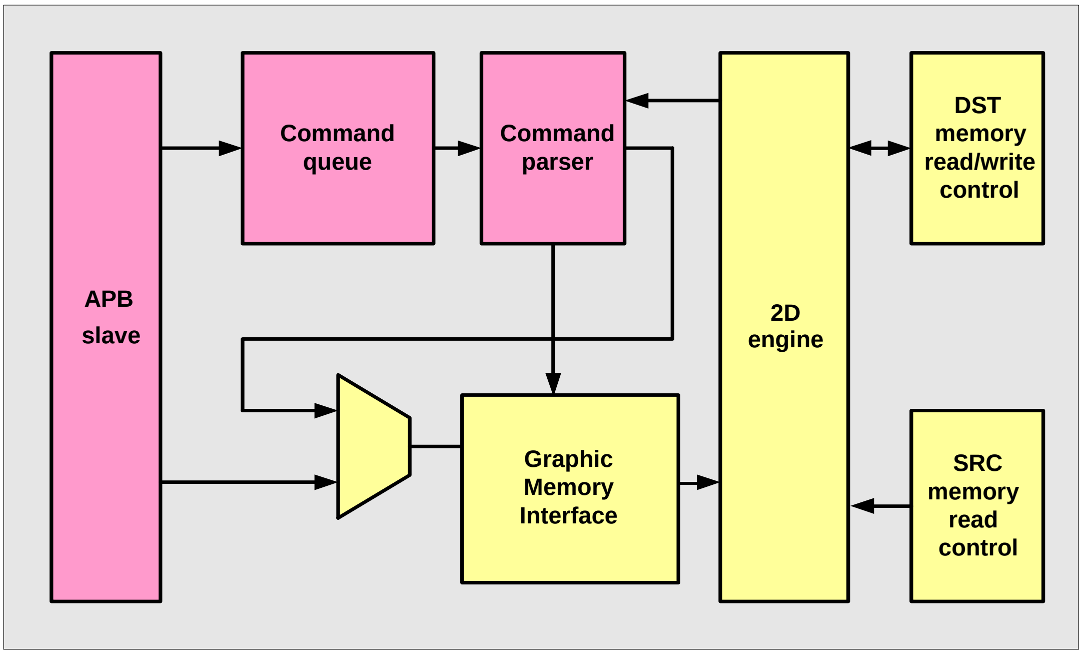


Figure 100 The command queue and 2D engine block diagram.

5.1.1.2 Features Introduction

5.1.1.2.1 2D Coordinate

The coordinates in the 2D engine are represented as 12-bit signed integers. The negative part is clipped during rendering. The maximum resolution can achieve 2047x2047 pixels. The programmed base address is mapped to the origin of the picture, which is illustrated in **Figure 101**.

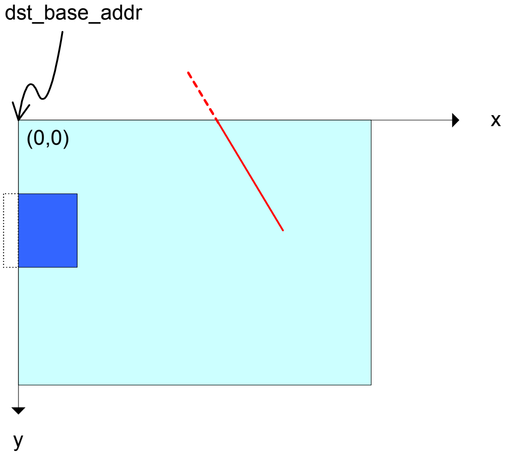


Figure 101 The coordinate of the 2D engine.

5.1.1.2.2 Color format

The 2D engine supports the color format of 8bpp, RGB565, RGB888, ARGB4444, and ARGB 8888. The color formats of source and destination can be specified separately. Note that when using the 8bpp format,

the source and destination color formats have to be the same, since table-lookup of color palette is not provided in 2D engine. Graphic modes of Bitblt, Bitblt with alpha blending, and Bitblt with binary ROP require color format setting for both source and destination. For other graphic modes, only destination color format needs to be specified. The possible settings are listed as **Table 87Table 88**.

Bitblt (Copy, ROP)	
Source color format	**Destination color format**
8bpp	8bpp
RGB565	RGB565
	RGB888
RGB888	RGB565
	RGB888
ARGB4444	ARGB4444
	ARGB8888
ARGB8888	ARGB4444
	ARGB8888

Table 87 source and destination color format setting for Bitblt.

Bitblt with Alpha Blending	
Source color format	**Destination color format**
8bpp	8bpp
RGB565	RGB565
	RGB888
RGB888	RGB565
	RGB888
ARGB4444	RGB565
	RGB888
ARGB8888	RGB565
	RGB888

Table 88 source and destination color format setting for alpha blending.

When source image is used, the source key function could be enabled or disabled. When enabled, the source color key is in the same format of source color. Be aware that the source key is still effective for alpha blending mode.

If the color format of source and destination is the same, no color conversion is necessary. But if source and destination have different color formats, the color values are converted between different formats as followed. If the source color format has less bit number than the destination, the source color value is shifted left to align with the most significant bit of the destination color. And the lower significant bits of destination color are filled with the most significant bits of source color. For example, to convert from ARGB4444 (a3a2a1a0r3r2r1r0g3g2g1g0b3b2b1b0) to ARGB8888, the value a3a2a1a0 a3a2a1a0r3r2r1r0 r3r2r1r0g3g2g1g0 g3g2g1g0b3b2b1b0 b3b2b1b0 is gotten. The conversion between RGB565 source and

RGB888 has another formula. A conversion factor is defined as followed: (2^d-1)/(2^s-1) in which d is the destination color format's bit number and s is the source's one. The destination color value is acquired by multiplying the source color value with the conversion factor and rounding to the nearest integer number. You can use WM_CPY_X control bit to select with conversion formula used when BitBlt. If the source color format has more bit number than the destination, the destination color value is acquired with just truncating the lower significant bits of the source color. For example, an ARGB8888 color value a7a6a5a4 a3a2a1a0r7r6r5r4r3r2r1r0g7g6g5g4g3g2g1g0b7b6b5b4b3b2b1b0 becomes r7r6r5r4r3g7g6g5g4g3g2b7b6b5b4b3 if the destination color format is RGB565.

5.1.1.2.3 Clipping Window

The setting for clipping window is effective for all the 2D graphics. A pair of minimum and maximum boundary is applied on destination side. The portion outside the clipping window will not be drawn to the destination, but the pixels on the boundary will be kept. The clipping operation is illustrated in **Figure 102**.

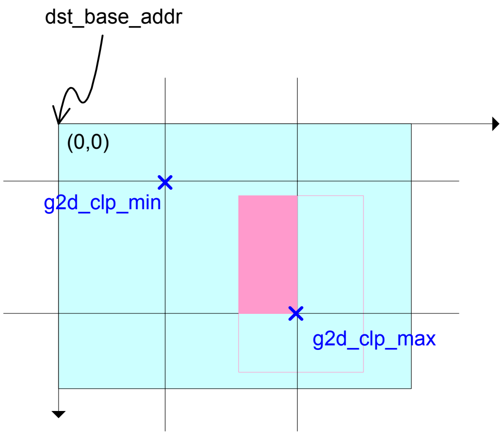


Figure 102 The clipping operation of the 2D engine.

To get more efficiency of BitBlt, or Font drawing, the source clipping can be enabled to avoid that the additional pixels of source, destination, pattern, or mask are fetched but are not written back because they are outside the destination clipping window. The setup procedure is as followed.

```
if (x_inc) {
        SRC_X = SRC_X;
        DST_X = DST_X;
        clp_diff_min_x = CLP_MIN_X – DST_X;
        if (clp_diff_min_x > 0) {
                SRC_CLP_X = src_x + (clp_diff_min_x % SRC_W);
                DST_CLP_X = CLP_MIN_X;
        }
        else {
                clp_diff_min_x = 0;
                SRC_CLP_X = SRC_X;
                DST_CLP_X = DST_X;
        }
}
else {
        SRC_X = SRC_X + SRC_W -1;
        DST_X = DST_X + DST_W – 1;
```

```
        clp_diff_min_x = DST_X – CLP_MAX_X;
        if (clp_diff_min_x > 0) {
                SRC_CLP_X = SRC_X – (clp_diff_min_x % SRC_W);
                DST_CLP_X = CLP_MAX_X;
        }
        else {
                clp_diff_min_x = 0;
                SRC_CLP_X = SRC_X;
                DST_CLP_X = DST_X;
        }
}
DST_CLP_W = DST_W – clp_diff_min_x;
DST_CLP_W = (DST_CLP_W <= 0)? 0 : DST_CLP_W;
```

You can replace the X's and W's above with Y's and H's respectively to get relative parameters about y-coordinate. And if the pattern or mask is used, PAT_CLP_XY or MSK_CLP_XY is derived in the same way. The source clipping is enabled with SRC_CLP_EN.

5.1.1.2.4 Bitblt operation

The Bitblt function copies the pixels from source picture to destination. To be more flexible, 4 copy directions and 7 kinds of rotations are provided when doing Bitblt operation. **Figure 103** illustrates the Bitblt operation and required settings.

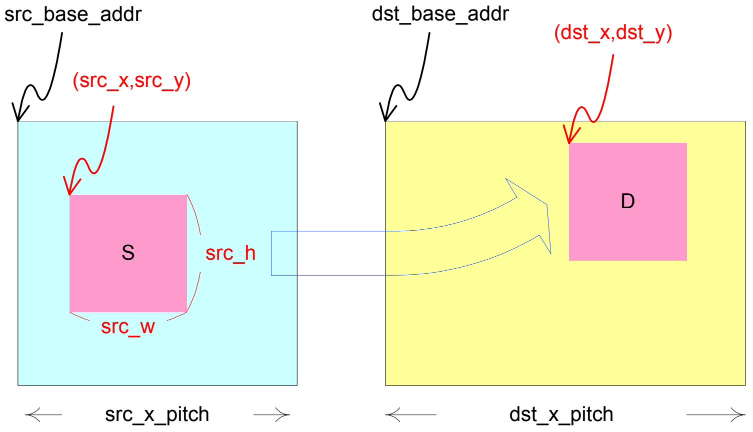


Figure 103 The clipping operation of the 2D engine.

Note that the size of source and destination blocks can be different. If the source block is larger than destination block, the size of destination block is used instead of the source size. When source block size is smaller than destination block size, the pattern of source block is repeated horizontally and vertically in the destination block, which is illustrated as **Figure 104** below.

 © 2009 MediaTek Inc.

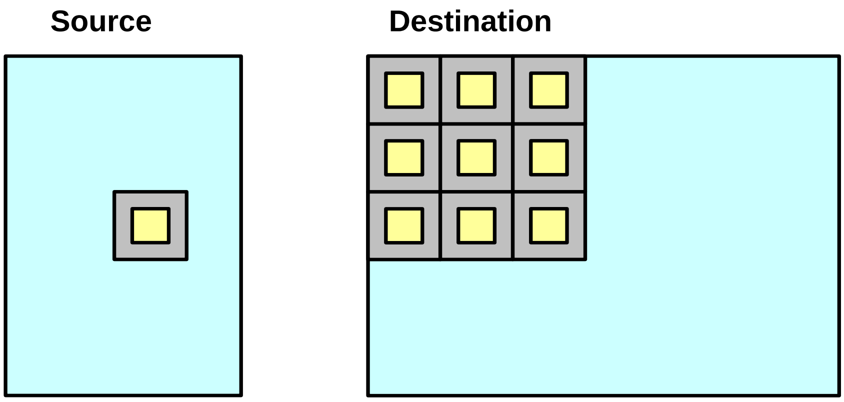


Figure 104 The Bitblt operation when destination size > source size.

5.1.1.2.4.1 Copy direction

When the source block and destination blocks are on the same picture, they may be overlapped by each other. To prevent error from occurring, 4 directions for Bitblt can be programmed. However, the copy direction shall not be enabled when doing rotation, or it will produce unwanted results. The 4 kinds of copy direction are shown in **Figure 105**.

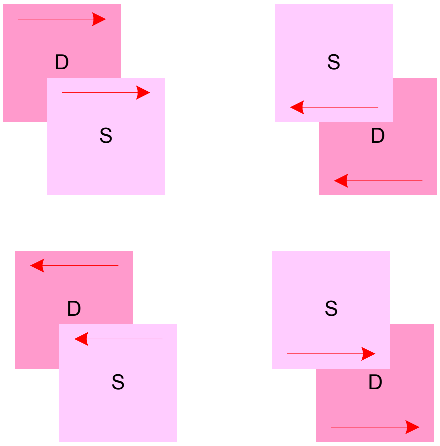


Figure 105 The 4 directions of Bitblt operation.

5.1.1.2.4.2 Rotation

To facilitate Bitblt operation, 7 kinds of rotation can be set at the same time. The rotation operation is illustrated as **Figure 106**. Here the rotation is done on the destination side, while the read sequence of pixels in source block is fixed.

 © 2009 MediaTek Inc.

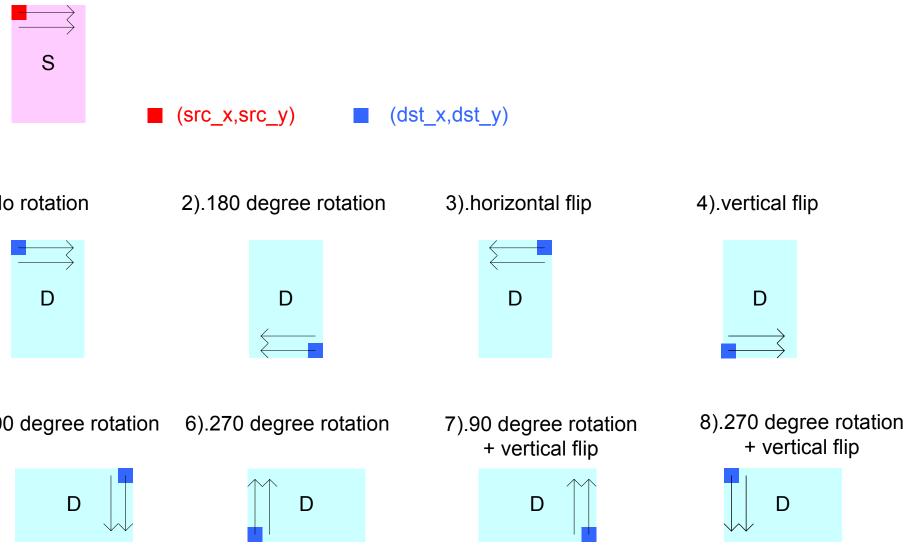


Figure 106 The rotations of Bitblt operation.

5.1.1.2.5 Bitblt with Alpha Blending

Similar to simple Bitblt operation, alpha blending function is provided as well. The pixels in source block are blended onto destination block. Blending is performed according the formula listed below:

$C = (alpha * Cs + (255 - alpha) * Cd)/255$,

where Cs is the source color, Cd is the destination color, and alpha is an unsigned integer range from 0 to 255.

The alpha value programmed into the 2D control registers is called constant alpha. When no alpha channel exists, the constant alpha is used to calculate blended color. If the alpha channel exists (in ARGB color mode), the per-pixel alpha is used for blending operation instead of constant alpha.

In addition, the setting of copy directions and rotations are also effective for alpha blending mode. Also, the size and color format of source block can be different from destination.

The alpha blending is approximated as followed in 2D accelerator:

x = (alpha * Cs) + (255 – alpha) * Cd

C = ((x + 128) >> 8 + (x + 128)) >> 8

The additional round in inner shift operation (that is, the red 128) can be enabled with WM_ALP_RND. And WM_CLR_X can control which color format being operated in the alpha blending pipeline. When WM_CLR_X is equal to ‘1’, all operands are converted to destination color format before the alpha blending operation is executed. Otherwise, all operands are converted to ARGB8888 color format before the alpha blending operation. And the results are converted to destination color format after the alpha blending operation.

5.1.1.2.6 Bitblt with ROP

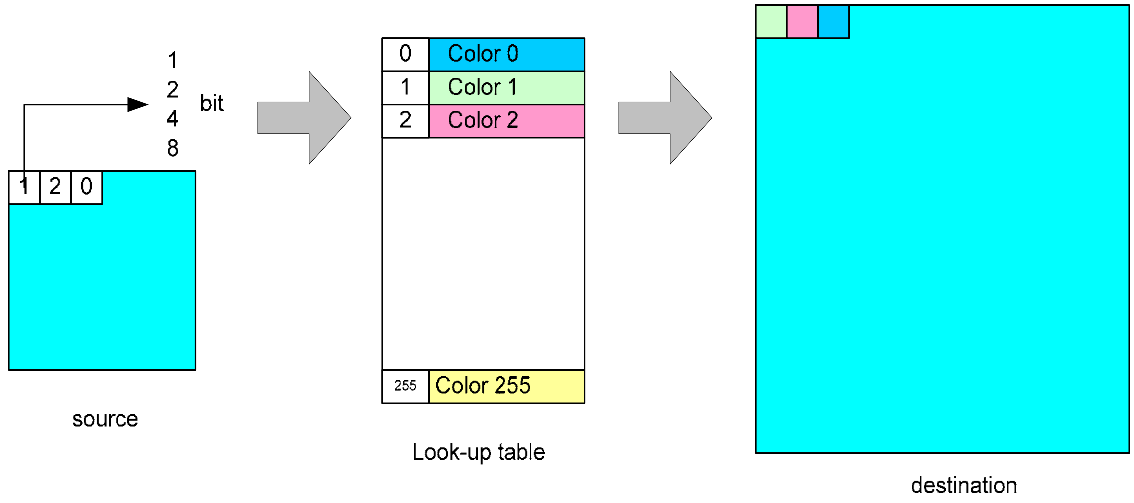
The ROP (Raster Operation) is another block-wise functional mode. Here the 2D engine provides a set of ROP2/ROP3/ROP4. The result of ROP3 (ternary ROP) is calculated by the bitwise logical operation of pixels of source, destination and pattern. The pattern is a maximum 64x64 block which could be programmed by G2D_PAT_BASE. The ROP code has 256 different combinations, which is listed in the definition of 2D control registers --- G2D_ROP_CODE. Please see sec.1.1.1.3 for detail descriptions. Note that disable

ROP3_EN in ROP mode can produce ROP2 (binary ROP) result. For ROP4, the result is one the calculated value of two ROP3 codes. The selection of ROP3 code depends on the value of the mask block which starting address is G2D_MSK_BASE.

Similar with other block-wise functions, the copy directions and rotations are also applicable in ROP mode. The size and color format of source and destination do not need to be the same.

5.1.1.2.7 Index color Bitblt

1/2/4/8 bit index color Bitblt is supported by 2D engine. The pixels in source block could be set as index color with 1, 2, 4, or 8 bits per pixel when doing Bitblt. A lookup table is used to find the 32bits ARGB color of the source pixel. The starting address of the index color table can be specified by setting G2D_BUF_STA_ADDR_0.



5.1.1.2.8 Scaled Bitblt

The scaling operation is supported by 2D engine. Simple repeat/drop algorithm is implemented for color key preservation. Maximum 63X scale up/down could be used with other Bitblt function, like alpha blending or copy direction selection.

5.1.1.2.9 Rectangle Fill with Color Gradient

Rectangle fill mode provides the configurations for color gradient for both x-direction and y-direction. Each of the color gradient of component A, R, G, B is represented by 9.16 signed fixed point number. In order to prevent color crossing the boundary of 0 and 255, it is clipped to 0 and 255 when performing gradient fill. When the color gradient is disabled, the rectangle is filled by one color. An example of gradient fill is shown in **Figure 107**.

© 2009 MediaTek Inc.

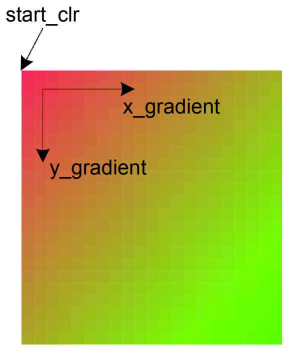


Figure 107 Rectangle gradient fill.

5.1.1.2.10 Font Drawing

The 2D engine helps to render fonts stored in one-bit-per-pixel format. It expends the zero bits to background color and expands one bits to foreground color. The background color can be set as transparent. The font drawing can be programmed as tilt, when given each line's tilt value.

The start bit of font drawing can be non-byte aligned to save memory usage for font caching. In addition, the rotations can be performed at the same time when drawing fonts.

5.1.1.2.11 Polygon Fill

In MT6516, 2D engine supports the function of polygon fill with its edges specified in memory. The maximum number of edge is 2047, which will occupy 32KB memory space (16 bytes per edge) during polygon fill processing. Software need to indicate the starting address of input edge list by setting G2D_BUF_STA_ADDR_0 and allocate another memory space for the polygon fill processing by setting G2D_BUF_STA_ADDR_1. It's noted that filling a polygon with a list of cross edges will cause an un-expected result. Dividing this kind of polygon into several ones without cross edges is recommended. Polygon fill with image is also supported. The maximum image pattern size is 64x64 which is needed to be put in memory starting from BUF_STA_ADDR_2.

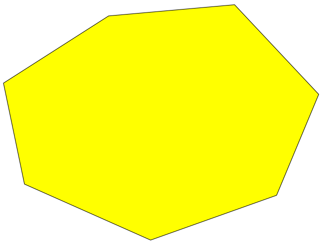


Figure 108 Polygon fill.

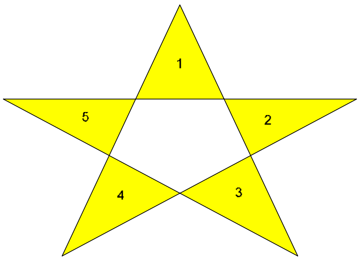


Figure 109 Cross edges. Divide into 1 - 5 triangles is recommended.

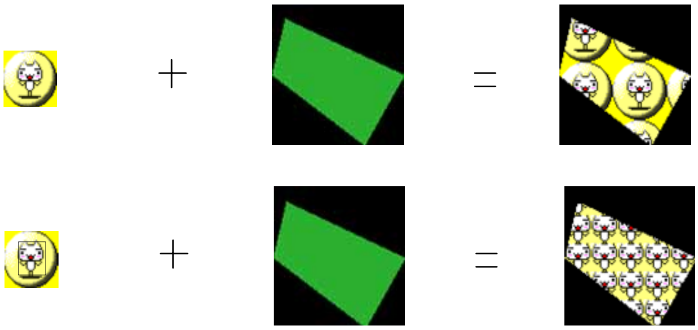


Figure 110 Polygon fill with image pattern.

5.1.1.2.12 G2D Engine Reset

When the G2D function is abnormal or the current executed G2D operation needs to be interrupted for some reason, the register RST in G2D_COM_CON could be used to reset the internal state machine in G2D engine. The setting of G2D configuration registers is unchanged. But at the same time when G2D engine is being reset, the read or write memory access transaction might not complete yet. This may cause G2D engine works abnormally. It is suggested that the TIDLE status bit in G2D_COM_STA needs to be checked before G2D engine is reset. That is, the completeness of R/W memory access transaction needs to be confirmed, then, the current executed G2D operation could be stopped.

5.1.1.3 Register Definitions

Table 1 The AP CCIF register mapping

summarizes the 2D engine register mapping on APB and through command queue. The base address of 2D engine is 80081000h.

APB Address	CMQ mapped Address	Register Function	Acronym
G2D+0100h	100h	2D engine fire mode control register	**FMODE_CON**
	102h	Reserved	
G2D+0104h	104h	2D Engine sub-mode control lower register	**SMODE_CON_L**
	106h	2D Engine sub-mode control higher register	**SMODE_CON_H**
G2D+0108h	108h	2D engine common control register	**COM_CON**
	10Ah	Reserved	
G2D+0110h	110h	2D engine status register	**STA**
	112h	Reserved	
G2D+0200h	200h	Source base address lower hword register	**SRC_BASE_L**
	202h	Source base address higher hword register	**SRC_BASE_H**
G2D+0204h	204h	Source pitch register	**SRC_PITCH**
	206h	Reserved	
G2D+0208h	208h	Source y coordinate register	**SRC_Y**

© 2009 MediaTek Inc.

	20Ah	Source x coordinate register	**SRC_X**
G2D+020Ch	20Ch	Source height register	**SRC_H**
	20Eh	Source width register	**SRC_W**
G2D+0210h	210h	Source color key lower hword register	**SRC_KEY_L**
	212h	Source color key higher hword register	**SRC_KEY_H**
G2D+0214h	214h	Destination avoid color lower hword register	**DST_AVO_CLR_L**
	216h	Destination avoid color higher hword register	**DST_AVO_CLR_H**
G2D+0218h	218h	Destination replacement color lower hword register	**DST_REP_CLR_L**
	21ah	Destination replacement color higher hword register	**DST_REP_CLR_H**
G2D+021Ch	21ch	Source y coordinate after source clipping register	**SRC_CLP_Y**
	21eh	Source x coordinate after source clipping register	**SRC_CLP_X**
G2D+0220h	220h	Destination height after source clipping register	**DST_CLP_H**
	222h	Destination width after source clipping register	**DST_CLP_W**
G2D+0300h	300h	Destination base address lower hword register	**DST_BASE_L**
	302h	Destination base address higher hword register	**DST_BASE_H**
G2D+0304h	304h	Destination Pitch Register	**DST_PITCH**
	306h	Reserved	
G2D+0308h	308h	Destination y coordinate register 0	**DST_Y0**
	30Ah	Destination x coordinate register 0	**DST_X0**
G2D+030Ch	30Ch	Destination y coordinate register 1	**DST_Y1**
	30Eh	Destination x coordinate register 1	**DST_X1**
G2D+0310h	310h	Destination y coordinate register 2	**DST_Y2**
	312h	Destination x coordinate register 2	**DST_X2**
G2D+0318h	318h	Destination height register	**DST_H**
	31Ah	Destination width register	**DST_W**
G2D+031Ch	31ch	Destination y coordinate after source clipping register	**DST_CLP_Y**
	31eh	Destination x coordinate after source clipping register	**DST_CLP_X**
G2D+0320h	320h	Pattern base address lower hword register	**PAT_BASE_L**
	322h	Pattern base address higher hword register	**PAT_BASE_H**
G2D+0324h	324h	Pattern pitch register	**PAT_PITCH**
	326h	Reserved	

© 2009 MediaTek Inc.

G2D+0328h	328h	Pattern width and high register	**PAT_WH**
	32Ah	Pattern x and y coordinate register	**PAT_XY**
G2D+032Ch	32Ch	Pattern offset of x/y coordinate register	**PAT_XYOFS**
	32eh	Pattern xy coordinate after source clipping register	**PAT_CLP_XY**
G2D+0330h	330h	Mask base address lower hword register	**MSK_BASE_L**
	332h	Mask base address high hword register	**MSK_BASE_H**
G2D+0334h	334h	Mask picth register	**MSK_PICTH**
	336h	Reserved	
G2D+0338h	338h	Mask y coordinate register	**MSK_Y**
	33Ah	Mask x coordinate register	**MSK_X**
G2D+033Ch	33Ch	Mask height register	**MSK_H**
	33Eh	Mask width register	**MSK_W**
G2D+0340h	340h	Resize down scale register	**RSZ_D**
	342h	Resize up scale register	**RSZ_U**
G2D+0344h	344h	ROP code0 and code1 register	**ROP_CODE**
	346h	Reserved	
G2D_348h	348h	Mask y coordinate after source clipping register	**MSK_CLP_Y**
	34ah	Mask x coordinate after source clipping register	**MSK_CLP_X**
G2D+400h	400h	Foreground color lower hword register	**FGCLR_L**
	402h	Foreground color high hword register	**FGCLR_H**
G2D+404h	404h	Background color lower hword register	**BGCLR_L**
	406h	Background color high hword register	**BGCLR_H**
G2D+408h	408h	Clipping minimum y coordinate register	**CLP_MIN_Y**
	40Ah	Clipping minimum x coordinate register	**CLP_MIN_X**
G2D+40Ch	40Ch	Clipping maximum y coordinate register	**CLP_MAX_Y**
	40Eh	Clipping maximum x coordinate register	**CLP_MAX_X**
G2D+410h	410h	Rectangle color gradient of alpha component x lower hword register	**ALPGR_X_L**
	412h	Rectangle color gradient of alpha component x higher hword register	**ALPGR_X_H**
G2D+414h	414h	Rectangle color gradient of red component x lower hword register	**REDGR_X_L**
	416h	Rectangle color gradient of red component x higher hword register	**REDGR_X_H**
G2D+418h	418h	Rectangle color gradient of green component x lower hword register	**GREENGR_X_L**
	41Ah	Rectangle color gradient of green component	**GREENGR_X_H**

		x higher hword register	
G2D+41Ch	41Ch	Rectangle color gradient of blue component x lower hword register	BLUEGR_X_L
	41Eh	Rectangle color gradient of blue component x higher hword register	BLUEGR_X_H
G2D+420h	420h	Rectangle color gradient of alpha component y lower hword register	ALPGR_Y_L
	422h	Rectangle color gradient of alpha component y higher hword register	ALPGR_Y_H
G2D+424h	424h	Rectangle color gradient of red component y lower hword register	REDGR_Y_L
	426h	Rectangle color gradient of red component y higher hword register	REDGR_Y_H
G2D+428h	428h	Rectangle color gradient of green component y lower hword register	GREENGR_Y_L
	42Ah	Rectangle color gradient of green component y higher hword register	GREENGR_Y_H
G2D+42Ch	42Ch	Rectangle color gradient of blue component y lower hword register	BLUEGR_Y_L
	42Eh	Rectangle color gradient of blue component y higher hword register	BLUEGR_Y_H
G2D+430h	430h	Buffer 0 start address lower hword register	BUF_STA_ADDR_0_L
	432h	Buffer 0 start address higher hword register	BUF_STA_ADDR_0_H
G2D+434h	434h	Buffer 1 start address lower hword register	BUF_STA_ADDR_1_L
	436h	Buffer 1 start address higher hword register	BUF_STA_ADDR_1_H
G2D+0700h ~ G2D+071Fh	700h ~ 71Fh		TILT_0300 ~ TILT_1F1C

Table 89 The 2D engine register mapping.

There are several function modes in 2D graphics engine. Some registers are shared between different them. **Table 90** summarizes the settings under different function modes.

APB Address	CMQ Address	Rectangle fill	Bitblt Operations	Font caching	Polygon Fill
G2D+0200h	200h		SRC_BASE	SRC_BASE	
G2D+0204h	204h		SRC_PITCH		
G2D+0208h	208h		SRC_XY		
G2D+020Ch	20Ch		SRC_SIZE		
G2D+0210h	210h		SRC_KEY	SRC_KEY	

 © 2009 MediaTek Inc.

G2D+214h	214h	**DST_AVO_CLR**	**DST_AVO_CLR**	**DST_AVO_CLR**	**DST_AVO_CLR**
G2D+218h	218h	**DST_REP_CLR**	**DST_REP_CLR**	**DST_REP_CLR**	**DST_REP_CLR**
G2D+0300h	300h	**DST_BASE**	**DST_BASE**	**DST_BASE**	**DST_BASE**
G2D+0304h	304h	**DST_PITCH**	**DST_PITCH**	**DST_PITCH**	**DST_PITCH**
G2D+0308h	308h	**DST_XY**	**DST_XY**	**DST_XY**	
G2D+030Ch	30Ch				
G2D+0310h	310h				
G2D+0318h	318h	**DST_SIZE**	**DST_SIZE**	**DST_SIZE**	
G2D+320h	320h		**PAT_BASE**		**PAT_BASE**
G2D+324h	324h		**PAT_PITCH**		**PAT_PITCH**
G2D+328h	328h		**PAT_XYWH**		**PAT_XYWH**
G2D+32Ch	32Ch		**PAT_XYOFS**		**PAT_XYOFS**
G2D+330h	330h		**MSK_BASE**		
G2D+334h	334h		**MSK_PITCH**		
G2D+338h	338h		**MSK_XY**		
G2D+33Ch	33Ch		**MSK_SIZE**		
G2D+340h	340h		**RSZ_UD**		
G2D+344h	344h		**ROP_CODE**		
G2D+0400h	400h	**START_CLR**		**FGCLR**	
G2D+0404h	404h		**DST_KEY**	**BGCLR**	
G2D+0408h	408h	**CLP_MIN**	**CLP_MIN**	**CLP_MIN**	**CLP_MIN**
G2D+040Ch	40Ch	**CLP_MAX**	**CLP_MAX**	**CLP_MAX**	**CLP_MAX**
G2D+0410h	410h	**ALPGD_X**			
G2D+0414h	414h	**RED_GD_X**			
G2D+0418h	418h	**GREEN_GD_X**			
G2D+041Ch	41Ch	**BLUE_GD_X**			
G2D+0420h	420h	**ALPGD_Y**			
G2D+0424h	424h	**RED_GD_Y**			
G2D+0428h	428h	**GREEN_GD_Y**			
G2D+042Ch	42Ch	**BLUE_GD_Y**			
G2D+0430h	430h		**BUF_ADDR_0**		**EDGE_ADDR**
G2D+0434h	434h		**BUF_ADDR_1**		**SORT_ADDR**
G2D+0700h ~ G2D+071Fh	700h ~ 71Fh	**TILT_0300 ~ TILT_1F1C**	**TILT_0300 ~ TILT_1F1C**	**TILT_0300 ~ TILT_1F1C**	

Table 90 2D engine common registers

Below shows common control registers.

 © 2009 MediaTek Inc.

G2D+0100h Graphic 2D Engine Fire Mode Control Register G2D_FMODE_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PAT_CLR_MODE				SRC_CLR_MODE				DST_CLR_MODE				G2D_ENG_MODE			
Type	R/W				R/W				R/W				R/W			
Reset	000				000				000				0000			

Write this register will fire the 2D engine according to the CLR_MODE and ENG_MODE field.

PAT_CLR_MODE pattern color mode

- **000** 8-bpp, LUT disabled
- **001** 16-bpp, RGB 565 format
- **010** 32-bpp, ARGB 8888 format
- **011** 24-bpp, RGB 888 format
- **101** 16-bpp, ARGB 4444 format

SRC_CLR_MODE source color mode

- **000** 8-bpp, LUT disabled
- **001** 16-bpp, RGB 565 format
- **010** 32-bpp, ARGB 8888 format
- **011** 24-bpp, RGB 888 format
- **101** 16-bpp, ARGB 4444 format
- **others** reserved

DST_CLR_MODE destination color mode

- **000** 8-bpp, LUT disabled
- **001** 16-bpp, RGB 565 format
- **010** 32-bpp, ARGB 8888 format
- **011** 24-bpp, RGB 888 format
- **101** 16-bpp, ARGB 4444 format
- **others** reserved

G2D_ENG_MODE 2D engine function mode

- **0110** Polygon fill.
- **1000** Rectangle fill.
- **1001** Bitblt.
- **1010** Bitblt with alpha blending.
- **1011** Bitblt with ROP.
- **1100** Font drawing.
- **others** not allowed

G2D+0104h Graphic 2D Engine Sub-mode Control Register G2D_SMODE_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FITA	FNBG	FMSB_FIRST	RSZ_EN	CLR_REP_EN		ROP4_EN	ROP3_EN	ALPHA							
Type	R/W	R/W	R/W	R/W	R/W			R/W	R/W				R/W			
Reset	0	0	0	0	0			0	0000				0000			

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		BIDX_EN	BIDX				LIMG		DST_KEY_EN	CLRGD_EN	BDIR		BITA	BROT		
Type		R/W	R/W						R/W	R/W	R/W		R/W	R/W		
Reset		0	0						0	0	11		0	111		

Write this register to set the 2D engine configuration.

FITA font italic enabled.
FNBG font drawing with no background color
FMSB_FIRST font drawing from most significant bit
RSZ_EN Scaled Bitblt mode enabled.
CLR_REP_EN Output color replacement enable.
ROP3_EN Ternary ROP enabled.
ALPHA Bit 7-0 of constant alpha value..
BIDX_EN Index color mode of Bitblt.
BIDX Index color selection of Bitblt, only effective when BIDX_EN is enabled.
- **00** 1 bit index color
- **01** 2 bit index color
- **10** 4 bit index color
- 11 8 bit index color

LIMG Polygon fill with image pattern. This function should not be enabled when clipping is enabled.
DST_KEY_EN Destination key enabled for Bitblt functions
CLRGR_EN Color gradient enabled for rectangle fill
BDIR Bitblt direction:
- **00** from lower right corner
- **01** from lower left corner
- **10** from upper right corner
- **11** from upper left corner

This field only takes effect when the Bitblt rotation is set as none (111). When doing rotation the Bitblt direction of source image is always from upper left corner.

BITA Bitblt italic enabled, using the tilt value defined in G2D_TILT_00 ~ G2D_TILT_1F registers. The tilt function should not be enabled in Alpha Blending and ROP mode.
BROT Bitblt rotation:
- **000** mirror then rotate 90
- **001** rotate 90
- **010** rotate 270
- **011** mirror then rotate 270
- **100** rotate 180
- **101** mirror
- **110** mirror then rotate 180
- 111 none

G2D+0108h Graphic 2D Engine Common Control Register G2D_COM_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIDLE_RST				WM_DST_XP	WM_PIT_U	WM_PMUL	WM_ALP_RND	WM_CPY_X	WM_CLR_X	SRC_CLP_EN	WR_BUF_EN	RD_BUF_EN	CLP_EN	SRCKEY_EN	RST
Type	R/W				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0				0	0	0	0	0	0	0	0	0	0	0	0

Write this register to set the 2D engine configuration.

RST 2D engine reset, only the state machine is reset, the content of control registers will not be reset.
SRCKEY_EN Source key enabled.
CLP_EN Destination clipping enabled.
RD_BUF_EN Read buffer enabled. Enable read buffer can improve read memory performance.
WR_BUF_EN Write buffer enabled. Enable write buffer could improve write memory performance.
SRC_CLP_EN Source clipping enabled.
WM_CLR_X Enable operation in destination color format with color format transform defined in WindowsMobile.
WM_CPY_X Enable operation in 32bpp color format with color format transform defined in WindowsMobile.
WM_ALP_RND Rounding before each shift operation in Alpha Blending.
WM_PMUL Pre-multiplied source format enabled. Only supported in ARGB8888.
WM_PIT_U Reserved.
WM_DST_XP Destination transparency enabled. Alpha blending operation on alpha channel when this bit is enabled.
TIDLE_RST Reset the R/W memory access transaction completeness status.

G2D+010Ch Graphic 2D Engine Interrupt Control Register G2D_IRQ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																EN
Type																R/W
Reset																0

Write this register to set the 2D engine IRQ configuration.

EN interrupt enable. The interrupt is negative edge sensitive.

G2D+0110h Graphic 2D Engine Common Status Register G2D_COM_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TIDLE	BUSY
Type															RO	RO
Reset															1	0

Read this register to get the 2D engine status. 2D engine may function abnormally if any 2D engine register is modified when BUSY.

BUSY 2D engine is busy
TIDLE Read this register to get the completeness of R/W memory access transaction. If the transaction is completed, this register will be asserted.

G2D+0200h Graphic 2D Source Base Address Register G2D_SRC_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRC_BASE[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC_BASE[15:0]															
Type	R/W															
Reset	0															

SRC_BASE The base address of source image.

G2D+0204h Graphic 2D Engine Source Pitch Register G2D_SRC_PITCH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					SRC_PITCH											
Type					R/W											
Reset					0											

SRC_PITCH The width of source image in the unit of pixels.

G2D+0208h Graphic 2D Engine Source X and Y Register G2D_SRC_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					SRC_X											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					SRC_Y											
Type					R/W											
Reset					0											

SRC_Y The starting y co-ordinate of source image. It must be positive although represented as 12-bit signed integer.
SRC_X The starting x co-ordinate of source image. It must be positive although represented as 12-bit signed integer.

G2D+020Ch Graphic 2D Engine Source Size Register G2D_SRC_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					SRC_W											
Type					R/W											
Reset					0											

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					SRC_H											
Type					R/W											
Reset					0											

SRC_H The source height for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.

SRC_W The source width for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.

G2D+0210h Graphic 2D Engine Source Color Key Register G2D_SRC_KEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRC_KEY[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC_KEY[15:0]															
Type	R/W															
Reset	0															

SRC_KEY The source color key. The color will be transparent if color keying is enabled.

G2D+0214h Graphic 2D Engine Destination Avoidance Color G2D_DST_AVO_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DST_AVO_CLR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_AVO_CLR[15:0]															
Type	R/W															
Reset	0															

DST_AVO_CLR The output color with DST_AVO_CLR would be replaced with DST_REP_CLR when CLR_REP_EN is enabled.

G2D+0218h Graphic 2D Engine Destination Replacement Color G2D_DST_REP_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DST_REP_CLR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_REP_CLR[15:0]															
Type	R/W															
Reset	0															

DST_REP_CLR The output color with DST_AVO_CLR would be replaced with DST_REP_CLR when CLR_REP_EN is enabled.

 © 2009 MediaTek Inc.

G2D+021Ch Graphic 2D Engine Source Clipping X and Y Register G2D_SRC_CLP_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					SRC_CLP_X											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					SRC_CLP_Y											
Type					R/W											
Reset					0											

SRC_CLP_X The starting value of source x co-ordinate in clipping window, unsigned 12-bit integer.
SRC_CLP_Y The starting value of source y co-ordinate in clipping window, unsigned 12-bit integer.

G2D+0220h Graphic 2D Engine Destination Size after Clipping Register G2D_DST_CLP_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					DST_CLP_W											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DST_CLP_H											
Type					R/W											
Reset					0											

DST_CLP_H The destination height for Bitblt, alpha blending and ROP after source clipping setup. It must be positive although represented as 12-bit unsigned integer.
DST_CLP_W The destination width for Bitblt, alpha blending and ROP after source clipping setup. It must be positive although represented as 12-bit unsigned integer.

G2D+0300h Graphic 2D Destination Base Address Register G2D_DST_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DST_BASE[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_BASE[15:0]															
Type	R/W															
Reset	0															

DST_BASE The base address of destination image.

G2D+0304h Graphic 2D Engine Destination Pitch Register G2D_DST_PITCH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DST_PITCH											
Type					R/W											
Reset					0											

DST_PITCH The width of destination image in the unit of pixels.

G2D+0308h Graphic 2D Engine Destination X and Y Register 0 G2D_DST_XY0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**DST_X0**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**DST_Y0**											
Type					R/W											
Reset					0											

(DST_X0 , DST_Y0) is used as the starting co-ordinate in Bitblt, alpha blending, ROP, and font drawing mode.

DST_X0 Represented by 12-bit signed integer. Negative co-ordinate is allowed.
DST_Y0 Represented by 12-bit signed integer. Negative co-ordinate is allowed.

G2D+0318h Graphic 2D Engine Destination Size Register G2D_DST_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**DST_W**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**DST_H**											
Type					R/W											
Reset					0											

DST_H The destination height for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.
DST_W The destination width for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.

G2D+031Ch Graphic 2D Engine Destination Clipping X and Y Register G2D_DST_CLP_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**DST_CLP_X**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**DST_CLP_Y**											
Type					R/W											
Reset					0											

SRC_CLP_X The starting value of destination x co-ordinate in clipping window, unsigned 12-bit integer.
SRC_CLP_Y The starting value of destination y co-ordinate in clipping window, unsigned 12-bit integer.

G2D+0320h Graphic 2D EnginePattern Base Register G2D_PAT_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PAT_BASE[31:16]**															
Type	R/W															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PAT_BASE[15:0]															
Type	R/W															
Reset	0															

PAT_BASE The base address of pattern image.

G2D+0324h Graphic 2D EnginePattern Pitch Register G2D_PAT_PITCH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PAT_PITCH					
Type											R/W					
Reset											0					

PAT_PITCH The width of pattern in the unit of pixels. The maximum width of pattern is 32.

G2D+0328h Graphic 2D Engine Pattern X Y W H Register G2D_PAT_XYWH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			PAT_X								PAT_Y					
Type			R/W								R/W					
Reset			0								0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PAT_W								PAT_H					
Type			R/W								R/W					
Reset			0								0					

PAT_X The starting x co-ordinate of pattern image for Ternary ROP or Polygon-Fill.
PAT_Y The starting y co-ordinate of pattern image for Ternary ROP or Polygon-Fill.
PAT_W The pattern width for Ternary ROP or Polygon-Fill.
PAT_H The pattern height for Ternary ROP or Polygon-Fill.

G2D+032Ch Graphic 2D Engine Pattern X Y Clipping and Offset Register G2D_PAT_XYOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			PAT_CLP_X								PAT_CLP_Y					
Type			R/W								R/W					
Reset			0								0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PAT_XOFS								PAT_YOFS					
Type			R/W								R/W					
Reset			0								0					

PAT_CLP_X The starting value of pattern x co-ordinate in clipping window, unsigned 6-bit integer.
PAT_CLP_Y The starting value of pattern y co-ordinate in clipping window, unsigned 6-bit integer.
PAT_XOFS The offset of x co-ordinate of pattern image for Ternary ROP or Polygon-Fill.

PAT_YOFS The offset of y co-ordinate of pattern image for Ternary ROP or Polygon-Fill.

G2D+0330h Graphic 2D EngineMask Base Register G2D_MSK_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MSK_BASE[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MSK_BASE[15:0]															
Type	R/W															
Reset	0															

MSK_BASE The base address of mask image.

G2D+0334h Graphic 2D Engine Mask Pitch Register G2D_MSK_PITCH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MSK_PITCH											
Type					R/W											
Reset					0											

SRC_PITCH The width of source image in the unit of pixels.

G2D+0338h Graphic 2D Engine Mask X and Y Register G2D_MSK_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					MSK_X											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MSK_Y											
Type					R/W											
Reset					0											

MSK_Y The starting y co-ordinate of mask image. It must be positive although represented as 12-bit signed integer.

MSK_X The starting x co-ordinate of mask image. It must be positive although represented as 12-bit signed integer.

G2D+033Ch Graphic 2D Engine Mask Size Register G2D_MSK_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					MSK_W											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MSK_H											
Type					R/W											
Reset					0											

MSK_H The mask height for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.

MSK_W The mask width for Bitblt, alpha blending and ROP. It must be positive although represented as 12-bit signed integer.

G2D+0340h Graphic 2D Engine Resize UP Donw Register G2D_RSZ_UD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											G2D_RSZ_UP					
Type											R/W					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											G2D_RSZ_DN					
Type											R/W					
Reset											0					

G2D_RSZ_UP The scaling up factor for Bitblt.
G2D_RSZ_DN The scaling down factor for Bitblt.

G2D+0344h Graphic 2D Engine ROP Code Register G2D_ROP_CODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ROP_CODE0								ROP_CODE1							
Type	R/W								R/W							
Reset	0								0							

ROP_CODE1 ROP3 code, or ROP4 code for mask bit is “1”.
ROP_CODE0 ROP4 code for mask bit is “0”.

Bitblt ROP Code	Boolean Function	Start Bit Position for Font Drawing
0000_0000	0 (Black)	Bit 0
0000_0001	~(S + D)	Bit 1
0000_0010	~S . D	Bit 2
0000_0011	~S	Bit 3
0000_0100	S . ~D	Bit 4
0000_0101	~D	Bit 5
0000_0110	S ^ D	Bit 6
0000_0111	~(S . D)	Bit 7
0000_1000	S . D	Bit 0
0000_1001	~(S ^ D)	Bit 1
0000_1010	D	Bit 2
0000_1011	~S + D	Bit 3
0000_1100	S	Bit 4
0000_1101	S + ~D	Bit 5

0000_1110	S + D	Bit 6
0000_1111	1 (White)	Bit 7

S = Source, D = Destination.

I = first quadrant, II = second quadrant, III = third quadrant, IV = fourth quadrant.

ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function
0000_0000	0 (Black)	0000_0001	~ ((P + S) + D)	0000_0010	(~ (P + S)) . D	0000_0011	~ (P + S)
0000_0100	(~ (P + D)) . S	0000_0101	~ (P + D)	0000_0110	~ ((~ (D ^ S)) + P)	0000_0111	~ ((D . S) + P)
0000_1000	(~ P) . D . S	0000_1001	~ ((D ^ S) + P)	0000_1010	(~ P) . D	0000_1011	~ (((~ D) . S) + P)
0000_1100	(~ P) . S	0000_1101	~ (((~ S) . D) + P)	0000_1110	~ ((~ (D + S)) + P)	0000_1111	~ P
0001_0000	(~ (D + S)) . P	0001_0001	~ (D + S)	0001_0010	~ ((~ (D ^ P)) + S)	0001_0011	~ ((D . P) + S)
0001_0100	~ ((~ (P ^ S)) + D)	0001_0101	~ ((P . S) + D)	0001_0110	(((~ (P . S)) . D) ^ S) ^ P	0001_0111	~ (((S ^ P) . (D ^ S)) ^ S)
0001_1000	(S ^ P) . (D ^ P)	0001_1001	~ (((~ (P . S)) . D) ^ S)	0001_1010	((S . P) + D) ^ P	0001_1011	~ (((P ^ S) . D) ^ S)
0001_1100	((D . P) + S) ^ P	0001_1101	~ (((P ^ D) . S) ^ D)	0001_1110	(D + S) ^ P	0001_1111	~ ((D + S) . P)
0010_0000	(~ S) . P . D	0010_0001	~ ((D ^ P) + S)	0010_0010	(~ S) . D	0010_0011	~ (((~ D) . P) + S)
0010_0100	(S ^ P) . (D ^ S)	0010_0101	~ (((~ (S . P)) . D) ^ P)	0010_0110	((P . S) + D) ^ S	0010_0111	((~ (P ^ S)) + D) ^ S
0010_1000	(P ^ S) . D	0010_1001	~ ((((P . S) + D) ^ S) ^ P)	0010_1010	(~ (P . S)) . D	0010_1011	~ (((S ^ P) . (P ^ D)) ^ S)
0010_1100	((D + S) . P) ^ S	0010_1101	((~ D) + S) ^ P	0010_1110	((D ^ P) + S) ^ P	0010_1111	~ (((~ D) + S) . P)
0011_0000	(~ S) . P	0011_0001	~ (((~ P) . D) + S)	0011_0010	((P + S) + D) ^ S	0011_0011	~ S
0011_0100	((D . S) + P) ^ S	0011_0101	((~ (D ^ S)) + P) ^ S	0011_0110	(D + P) ^ S	0011_0111	~ ((D + P) . S)
0011_1000	((D + P) . S) ^ P	0011_1001	((~ D) + P) ^ S	0011_1010	((D ^ S) + P) ^ S	0011_1011	~ (((~ D) + P) . S)
0011_1100	P ^ S	0011_1101	((~ (D + S))	0011_1110	(((~ S) . D)	0011_1111	~ (P . S)

ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function
			+ P) ^ S		+ P) ^ S		
0100_0000	(~ D) . S . P	0100_0001	~ ((P ^ S) + D)	0100_0010	(S ^ D) . (P ^ D)	0100_0011	~ (((~ (D . S)) . P) ^ S)
0100_0100	(~ D) . S	0100_0101	~ (((~ S) . P) + D)	0100_0110	((P . D) + S) ^ D	0100_0111	~ (((D ^ P) . S) ^ P)
0100_1000	(D ^ P) . S	0100_1001	~ (((P . D) + S) ^ D ^ P)	0100_1010	((S + D) . P) ^ D	0100_1011	((~ S) + D) ^ P
0100_1100	(~ (D . P)) . S	0100_1101	~ (((S ^ P) + (S ^ D)) ^ S)	0100_1110	((S ^ P) + D) ^ P	0100_1111	~ (((~ S) + D) . P)
0101_0000	(~ D) . P	0101_0001	~ (((~ P) . S) + D)	0101_0010	((S . D) + P) ^ D	0101_0011	~ (((D ^ S) . P) ^ S)
0101_0100	~ ((~ (P + S)) + D)	0101_0101	~ D	0101_0110	(P + S) ^ D	0101_0111	~ ((P + S) . D)
0101_1000	((S + P) . D) ^ P	0101_1001	((~ S) + P) ^ D	0101_1010	D ^ P	0101_1011	((~ (S + D)) + P) ^ D
0101_1100	((S ^ D) + P) ^ D	0101_1101	~ (((~ S) + P) . D)	0101_1110	(((~ D) . S) + P) ^ D	0101_1111	~ (D . P)
0110_0000	(D ^ S) . P	0110_0001	~ (((D . S) + P) ^ S ^ D)	0110_0010	((P + D) . S) ^ D	0110_0011	((~ P) + D) ^ S
0110_0100	((P + S) . D) ^ S	0110_0101	((~ P) + S) ^ D	0110_0110	D ^ S	0110_0111	((~ (P + S)) + D) ^ S
0110_1000	~ (((~ (D + S)) + P) ^ S ^ D)	0110_1001	~ (D ^ S ^ P)	0110_1010	(P . S) ^ D	0110_1011	~ (((P + S) . D) ^ S ^ P)
0110_1100	(D . P) ^ S	0110_1101	~ (((P + D) . S) ^ D ^ P)	0110_1110	(((~ S) + P) . D) ^ S	0110_1111	~ ((~ (D ^ S)) . P)
0111_0000	(~ (D . S)) . P	0111_0001	~ (((S ^ D) . (P ^ D)) ^ S)	0111_0010	((P ^ S) + D) ^ S	0111_0011	~ (((~ P) + D) . S)
0111_0100	((P ^ D) + S) ^ D	0111_0101	~ (((~ P) + S) . D)	0111_0110	(((~ S) . P) + D) ^ S	0111_0111	~ (S . D)
0111_1000	(D . S) ^ P	0111_1001	~ (((D + S) . P) ^ S ^ D)	0111_1010	(((~ D) + S) . P) ^ D	0111_1011	~ ((~ (D ^ P)) . S)

ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function
0111_1100	(((~ S) + D) . P) ^ S	0111_1101	~ ((~ (P ^ S)) . D)	0111_1110	(S ^ P) + (S ^ D)	0111_1111	~ (P . S . D)
1000_0000	P . S . D	1000_0001	~ ((S ^ P) + (S ^ D))	1000_0010	(~ (P ^ S)) . D	1000_0011	~ ((((~ S) + D) . P) ^ S)
1000_0100	(~ (D ^ P)) . S	1000_0101	~ ((((~ P) + S) . D) ^ P)	1000_0110	((D + S) . P) ^ S ^ D	1000_0111	~ ((D . S) ^ P)
1000_1000	D . S	1000_1001	~ ((((~ S) . P) + D) ^ S)	1000_1010	((~ P) + S) . D	1000_1011	~ (((P ^ D) + S) ^ D)
1000_1100	((~ P) + D) . S	1000_1101	~ (((P ^ S) + D) ^ S)	1000_1110	((S ^ D) . (P ^ D)) ^ S	1000_1111	~ ((~ (D . S)) . P)
1001_0000	(~ (D ^ S)) . P	1001_0001	~ ((((~ S) + P) . D) ^ S)	1001_0010	((D + P) . S) ^ P ^ D	1001_0011	~ ((P . D) ^ S)
1001_0100	((P + S) . D) ^ S ^ P	1001_0101	~ ((P . S) ^ D)	1001_0110	P ^ S ^ D	1001_0111	((~ (P + S)) + D) ^ S ^ P
1001_1000	~ (((~ (P + S)) + D) ^ S)	1001_1001	~ (D ^ S)	1001_1010	((~ S) . P) ^ D	1001_1011	~ (((P + S) . D) ^ S)
1001_1100	((~ D) . P) ^ S	1001_1101	~ (((P + D) . S) ^ D)	1001_1110	((D . S) + P) ^ S ^ D	1001_1111	~ ((D ^ S) . P)
1010_0000	D . P	1010_0001	~ ((((~ P) . S) + D) ^ P)	1010_0010	((~ S) + P) . D	1010_0011	~ (((S ^ D) + P) ^ D)
1010_0100	~ (((~ (S + P)) + D) ^ P)	1010_0101	~ (P ^ D)	1010_0110	((~ P) . S) ^ D	1010_0111	~ (((P + S) . D) ^ P)
1010_1000	(P + S) . D	1010_1001	~ ((P + S) ^ D)	1010_1010	D	1010_1011	(~ (P + S)) + D
1010_1100	((D ^ S) . P) ^ S	1010_1101	~ (((S . D) + P) ^ D)	1010_1110	((~ P) . S) + D	1010_1111	(~ P) + D
1011_0000	((~ S) + D) . P	1011_0001	~ (((S ^ P) + D) ^ P)	1011_0010	((S ^ P) + (S ^ D)) ^ S	1011_0011	~ ((~ (D . P)) . S)
1011_0100	((~ D) . S) ^ P	1011_0101	~ (((S + D) . P) ^ D)	1011_0110	((D . P) + S) ^ P ^ D	1011_0111	~ ((D ^ P) . S)
1011_1000	((D ^ P) . S) ^ P	1011_1001	~ (((P . D) + S) ^ D)	1011_1010	((~ S) . P) + D	1011_1011	(~ S) + D

ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function	ROP code	Boolean Function
1011_1100	((~ (D . S)) . P) ^ S	1011_1101	~ ((S ^ D) . (P ^ D))	1011_1110	(P ^ S) + D	1011_1111	(~ (P . S)) + D
1100_0000	P . S	1100_0001	~ ((((~ S) . D) + P) ^ S)	1100_0010	~ (((~ (D + S)) + P) ^ S)	1100_0011	~ (P ^ S)
1100_0100	((~ D) + P) . S	1100_0101	~ (((D ^ S) + P) ^ S)	1100_0110	((~ P) . D) ^ S	1100_0111	~ (((D + P) . S) ^ P)
1100_1000	(D + P) . S	1100_1001	~ ((P + D) ^ S)	1100_1010	((S ^ D) . P) ^ D	1100_1011	~ (((D . S) + P) ^ S)
1100_1100	S	1100_1101	(~ (D + P)) + S	1100_1110	((~ P) . D) + S	1100_1111	(~ P) + S
1101_0000	((~ D) + S) . P	1101_0001	~ (((D ^ P) + S) ^ P)	1101_0010	((~ S) . D) ^ P	1101_0011	~ (((D + S) . P) ^ S)
1101_0100	((S ^ P) . (D ^ P)) ^ S	1101_0101	~ ((~ (P . S)) . D)	1101_0110	((P . S) + D) ^ S ^ P	1101_0111	~ ((P ^ S) . D)
1101_1000	((S ^ P) . D) ^ P	1101_1001	~ (((P . S) + D) ^ S)	1101_1010	((~ (S . D)) . P) ^ D	1101_1011	~ ((S ^ P) . (S ^ D))
1101_1100	((~ D) . P) + S	1101_1101	(~ D) + S	1101_1110	(D ^ P) + S	1101_1111	(~ (D . P)) + S
1110_0000	(D + S) . P	1110_0001	~ ((D + S) ^ P)	1110_0010	((P ^ D) . S) ^ D	1110_0011	~ (((D . P) + S) ^ P)
1110_0100	((P ^ S) . D) ^ S	1110_0101	~ (((S . P) + D) ^ P)	1110_0110	((~ (P . S)) . D) ^ S	1110_0111	~ ((S ^ P) . (D ^ P))
1110_1000	((S ^ P) . (D ^ S)) ^ S	1110_1001	~ (((~(D . S)) . P) ^ S ^ D)	1110_1010	(P . S) + D	1110_1011	(~ (P ^ S)) + D
1110_1100	(D . P) + S	1110_1101	(~ (D ^ P)) + S	1110_1110	D + S	1110_1111	(~ P) + D + S
1111_0000	P	1111_0001	(~ (D + S)) + P	1111_0010	((~ S) . D) + P	1111_0011	(~ S) + P
1111_0100	((~ D) . S) + P	1111_0101	(~ D) + P	1111_0110	(D ^ S) + P	1111_0111	(~ (D . S)) + P
1111_1000	(D . S) + P	1111_1001	(~ (D ^ S)) + P	1111_1010	D + P	1111_1011	(~ S) + P + D
1111_1100	P + S	1111_1101	(~ D) + S + P	1111_1110	P + S + D	1111_1111	1 (Wite)

S = Source, D = Destination, P=Pattern.

© 2009 MediaTek Inc.

G2D+0348h Graphic 2D Engine Mask Clipping X and Y Register G2D_MSK_CLP_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					MSK_CLP_X											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MSK_CLP_Y											
Type					R/W											
Reset					0											

MSK_CLP_X The starting value of mask x co-ordinate in clipping window, unsigned 12-bit integer.

MSK_CLP_Y The starting value of mask y co-ordinate in clipping window, unsigned 12-bit integer.

G2D+0400h Graphic 2D Engine Foreground Color Register G2D_ FGCLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FGCLR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FGCLR[15:0]															
Type	R/W															
Reset	0															

FGCLR The foreground color used for font drawing. It is also the start color of rectangle fill. The format of foreground color depends on the source color mode set in G2D_FMODE_CON register.

G2D+0404h Graphic 2D Engine Background Color Register G2D_BGCLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BGCLR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BGCLR[15:0]															
Type	R/W															
Reset	0															

BGCLR The background color of the source. The format of background color depends on the source color mode set in G2D_FMODE_CON register.

G2D+0408h Graphic 2D Engine Clipping Minimum Register G2D_CLIP_MIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						CLIP_MIN_X										
Type						R/W										
Reset						0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						CLIP_MIN_Y										
Type						R/W										
Reset						0										

CLIP_MIN_X The minimum value of x co-ordinate in clipping window, signed 12-bit integer.

CLIP_MIN_Y The minimum value of y co-ordinate in clipping window, signed 12-bit integer..

G2D+040ch Graphic 2D Engine Clipping Maximum Register G2D_CLIP_MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						CLIP_MAX_X										
Type						R/W										
Reset						11111111111										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						CLIP_MAX_Y										
Type						R/W										
Reset						11111111111										

CLIP_MAX_X The maximum value of x co-ordinate in clipping window, signed 12-bit integer...

CLIP_MAX_Y The maximum value of y co-ordinate in clipping window, signed 12-bit integer..

G2D+0410h Graphic 2D X Alpha Gradient Register G2D_ALPGR_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								ALPHA_GR_X[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA_GR_X[15:0]															
Type	R/W															
Reset	0															

The color gradient of alpha in x direction for rectangle gradient fill.

ALPHA_GR_X The color gradient of alpha channel, represented in signed 9.16 format.

G2D+0414h Graphic 2D X Red Gradient Register G2D_REDGR_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								RED_GR_X[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RED_GR_X[15:0]															
Type	R/W															
Reset	0															

The color gradient of red in x direction for rectangle gradient fill.

RED_GR_X The color gradient of red component, represented in signed 9.16 format.

G2D+0418h Graphic 2D X Green Gradient Register G2D_GREENGR_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								GREEN_GR_X[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN_GR_X[15:0]															
Type	R/W															

Reset	0

GREEN_GR_X The color gradient of blue component, represented in signed 9.16 format.

G2D+041Ch Graphic 2D X Blue Gradient Register G2D_BLUEGR_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								BLUE_GR_X[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLUE_GR_X[15:0]															
Type	R/W															
Reset	0															

BLUE_GR_X The color gradient of blue component, represented in signed 9.16 format.

G2D+0420h Graphic 2D Y Alpha Gradient Register G2D_ALPGR_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								ALPHA_GR_Y[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA_GR_Y[15:0]															
Type	R/W															
Reset	0															

The color gradient of alpha in x direction for rectangle gradient fill.

ALPHA_GR_Y The color gradient of alpha channel, represented in signed 9.16 format.

G2D+0424h Graphic 2D Y Red Gradient Register G2D_REDGR_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								RED_GR_Y[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RED_GR_Y[15:0]															
Type	R/W															
Reset	0															

The color gradient of red in x direction for rectangle gradient fill.

RED_GR_Y The color gradient of red component, represented in signed 9.16 format.

G2D+0428h Graphic 2D Y Green Gradient Register G2D_GREENGR_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								GREEN_GR_Y24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GREEN_GR_Y15:0]
Type	R/W
Reset	0

GREEN_GR_Y The color gradient of blue component, represented in signed 9.16 format.

G2D+042Ch Graphic 2D Y Blue Gradient Register G2D_BLUEGR_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								BLUE_GR_Y[24:16]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLUE_GR_Y[15:0]															
Type	R/W															
Reset	0															

BLUE_GR_Y The color gradient of blue component, represented in signed 9.16 format.

G2D+0430h Graphic 2D Engine Buffer Start Address 0 G2D_BUF_STA_ADDR_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF_STA_ADDR_0[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF_STA_ADDR_0[15:0]															
Type	R/W															
Reset	0															

BUF_STA_ADDR_0 The buffer 0 start address. Buffer 0 is used for raw edge data for Polygon-Fill function, index color table for Index-color Bitblt mode.

G2D+0434h Graphic 2D Engine Buffer Start Address 1 G2D_BUF_STA_ADDR_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF_STA_ADDR_1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF_STA_ADDR_1[15:0]															
Type	R/W															
Reset	0															

BUF_STA_ADDR_1 The buffer 1 start address. Buffer 1 is used for storing edge processing temporal data for Polygon-Fill function.

5.1.2 Command Queue

5.1.2.1 General Description

To enhance MMI display and gaming experiences, a command queue controllrt is implemented for further offloading of MCU. If the command queue is enabled, software program has to check the command queue

free space before writing to the command queue data register. Command queue parser will consume command queue entries upon 2D engine requests. **Figure 100** shows the command queue and 2D engine block diagram.

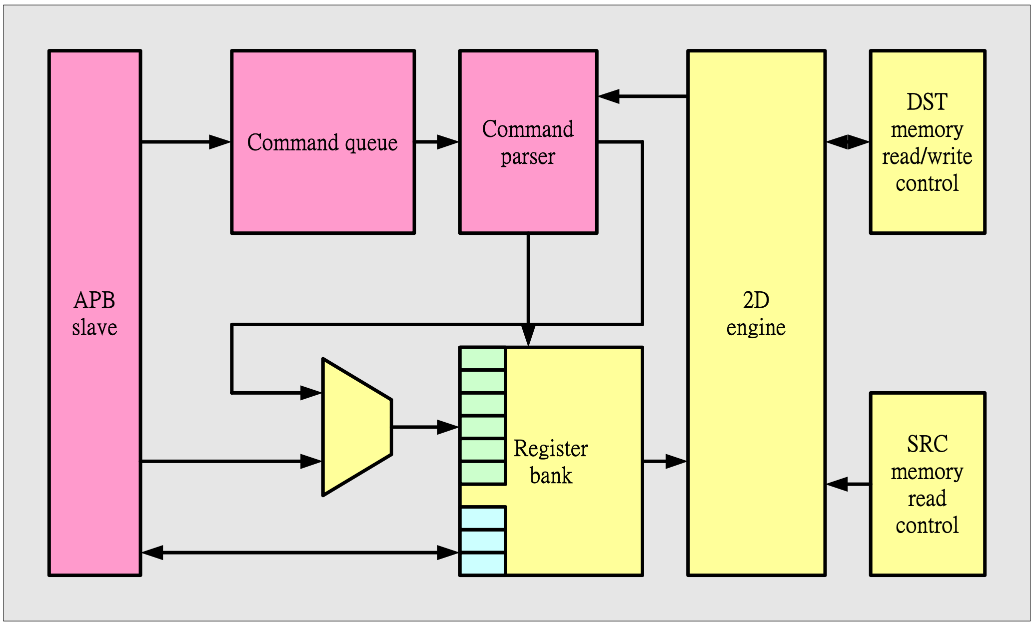


Figure 111 The command queue and 2D engine block diagram.

5.1.2.2 Register Definitions

MCU APB bus registers are listed as followings. The base address of the command queue controller is 80082000h.

GCMQ+0000h Graphic Command Queue Control Register GCMQ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TIDLE_RST	WOL	EN
Type														R/W	R/W	R/W
Reset														0	0	0

EN command queue enable. When EN is low, the command queue controller will be reset.

WOL avoid reading when writing enable. When WOL is high, prevent command queue reading command when MCU is writing command queue.

TIDLE_RST Reset the R/W memory access transaction completeness status.

GCMQ+0004h Graphic Command Queue Status Register GCMQ_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														TIDLE	EMPTY	FULL
Type														RO	RO	RO

Reset														1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**FREE**										
Type						RO										
Reset						100000000										

FREE number of free command queue entries

FULL Command queue full

EMPTY Command queue empty

TIDLE Read this register to get the completeness of R/W memory access transaction. If the transaction is completed, this register will be asserted.

GCMQ+0008h Graphic Command Queue Data Register GCMQ_DAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**ADDR**											
Type					WO											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA**															
Type	WO															

ADDR [11:0] write address for mapped 2D engine registers

DATA [15:0] write data for mapped 2D engine registers

GCMQ+000Ch Graphic Command Queue Base Address Register GCMQ_BASE_ADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BASE_ADD[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BASE_ADD[15:0]**															
Type	R/W															

BASE_ADD the starting address of the command queue in the memory.

Note : This field only can be modified while the command queue is not enabled. Otherwise the behavior of the command queue will be unpredictable.

GCMQ+0010h Graphic Command Queue Buffer Length Register GCMQ_LENGTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**LENGTH**										
Type						R/W										

LENGTH[9:0] the occupied space of the command queue in the memory is LENGTH *4Bytes.

Note : This field only can be modified while the command queue is not enabled. Otherwise the behavior of the command queue will be unpredictable.

GCMQ+0014h Graphic Command Queue Current Register GCMQ_DMA_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GCMQ_DMA_ADDR															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GCMQ_DMA_ADDR															
Type	RO															

GCMQ_DMA_ADDR the current read or write DMA address of GCMQ.

5.2 Audio SRC Mixer

5.2.1 General Description

Audio SRC (Sample Rate Conversion) Mixer (ASM) is am MCU coprocessor to convert the audio data from one sampling rate to another and mixed different sampled data together as shown in **Figure 112** Architecture of audio SRC mixer.

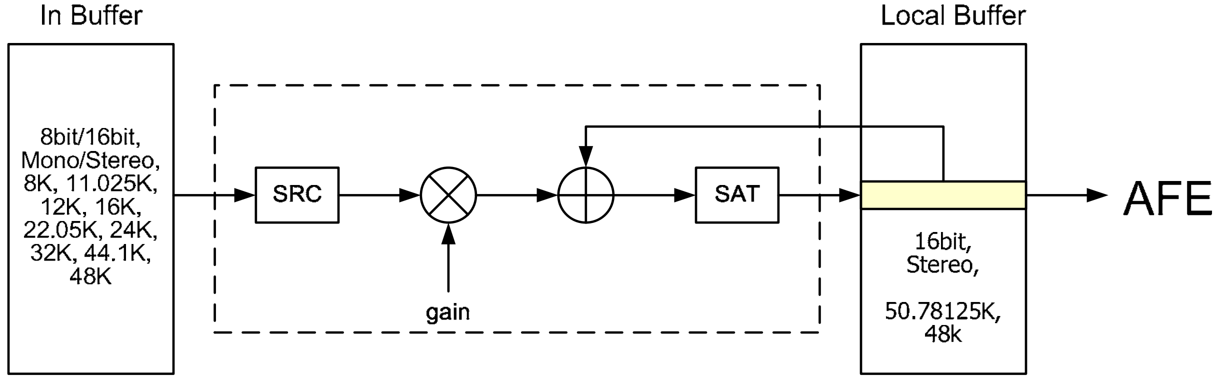


Figure 112 ***Architecture of audio SRC mixer***

The input sampling rate of PCM data may from 8k to 48k, mono or stereo, 8 bit or 16 bit, while the output data are at the sampling rate of 48k or 50.78125k (6.5MHz / 128) which is controlled by MCU. Moreover, a gain stage is implemented to support gain control for converted signals.

The implementation of SRC is to up-sample the input data to 16X sampling rate and followed by a third-order interpolator which converts the data to the desired rate (48 kHz or 50.78125 kHz).

5.2.2 Register Definitions

The Base Address of ASM is 0x0x8008F000

5.2.2.1 Audio SRC Mixer

MCU APB bus registers in audio SRC mixer are listed as follows.

 © 2009 MediaTek Inc.

0x8008F000 ASM PCM Input Ring Buffer Base Register ASM_IBUF_BASE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IBUF_BASE0[31:16]															
Type	R/W															
Reset	0000_0000_0000_0000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IBUF_BASE0[15:0]															
Type	R/W															
Reset	0000_0000_0000_0000															

The register indicates the starting address of the input PCM data for ASM. Please always set IBUF_BASE[3:0] = 4'h0 for the convenience of the hardware implementation.

IBUF_BASE0 Address of the starting point for the PCM input buffer.

Register Address	Register Function	Acronym
0x8008F020	ASM PCM Input Ring Buffer Base Register for Block 1	ASM_IBUF_BASE 1
0x8008F040	ASM PCM Input Ring Buffer Base Register for Block 2	ASM_IBUF_BASE 2

Table 91 ***PCM Input Ring Buffer Base Registers***

0x8008F004 ASM PCM Input Ring Buffer End Register ASM_IBUF_END0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IBUF_END0[31:16]															
Type	R/W															
Reset	0000_0000_0000_0000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IBUF_END0[15:0]															
Type	R/W															
Reset	0000_0000_0000_0000															

The register indicates the end address of the PCM ring buffer. The cursor for the PCM ring buffer will go back to the base address whenever the buffer-end is reached. Please always set IBUF_END0[3:0] = 4'hf for the convenience of the hardware implementation.

IBUF_END0 End address of the PCM input ring buffer.

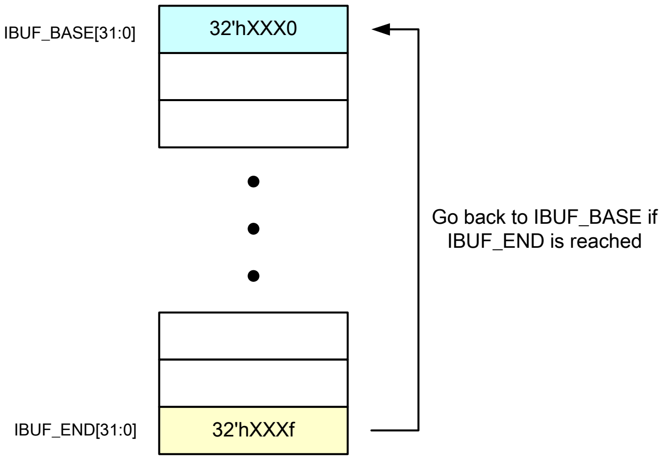


***Figure 113** Example of Input Buffer Address*

Register Address	Register Function	Acronym
0x8008F024	ASM PCM Input Ring Buffer End Register for Block 1	ASM_IBUF_END1
0x8008F044	ASM PCM Input Ring Buffer End Register for Block 2	ASM_IBUF_END2

Table 92 ***PCM Input Ring Buffer End Registers***

0x8008F008 ASM PCM Input Ring Buffer Cursor Register ASM_IBUF_CUR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IBUF_CUR0[31:16]**															
Type	RO															
Reset	-															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IBUF_CUR0[15:0]**															
Type	RO															
Reset	-															

This register indicates the address of the next PCM input ring buffer.

IBUF_CUR0 Position of the PCM input ring buffer.

Register Address	Register Function	Acronym
0x8008F028	ASM PCM Input Ring Buffer Cursor Register for Block 1	ASM_IBUF_CUR1
0x8008F048	ASM PCM Input Ring Buffer Cursor Register for Block 2	ASM_IBUF_CUR2

© 2009 MediaTek Inc.

Table 93 ***PCM Input Ring Buffer Cursor Registers***

0x8008F100 ASM MCU Control Register 0 ASM_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										BLK2_ON	BLK1_ON	BLK0_ON				ASMON
Type										R/W	R/W	R/W				R/W
Reset										0	0	0				0

This register is set to start ASM operation. Make sure there are at least one of BLK0_ON to BLK2_ON is raised while ASM is asserted. Never close all blocks if ASMON is not released.

ASMON Set this register to start the operation of the ASM. All other registers should be set to the desired values before this register is asserted.

0 Off

1 On

BLK0_ON Switch of block 0.

0 off

1 on

BLK1_ON Switch of block 1.

0 off

1 on

BLK2_ON Switch of block 2.

0 off

1 on

0x8008F104 ASM MCU Control Register 1 ASM_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															CLK_SWITCH_EN	OFS
Type															R/W	R/W
Reset															1	0

This register sets the configuration of ASM.

CLK_SWITCH_EN Enable signal for bus clock to speed down while ASM is On.

0 Disable. Bus clock should be 104MHz while ASM is on.

1 Enable

OFS Output sampling rate of the ASM.

0 50.78125 kHz (To AFE)

1 48 kHz (To I2S)

0x8008F010 ASM MCU Setting Register for Block 0 ASM_SET0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										IBIT0	IMODE0		IFS0			
Type										R/W	R/W		R/W			
Reset										0	0		0000			

This register is the I/O settings of the ASM for block 0.

IBIT0 Bit number of the input data for block 0.

0 8 bit

1 16 bit

IMODE0 Input Data Mode for block 0.

0 Stereo

1 Mono

IFS0 Input sampling rate of the ASM for block 0.

0000 48 kHz

0001 44.1 kHz

0010 32 kHz

0011 24 kHz

0100 22.05 kHz

0101 16 kHz

0110 12 kHz

0111 11.025 kHz

1000 8 kHz

Others 48 kHz

Register Address	Register Function	Acronym
0x8008F030	ASM MCU Setting Register for Block 1	ASM_SET1
0x8008F050	ASM MCU Setting Register for Block 2	ASM_SET2

Table 94 ***Block Setting Registers***

0x8008F014 ASM Gain Setting Register for Block 0 ASM_GAIN0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GAIN0															
Type	R/W															
Reset	0001_0000_0000_0000															

This register sets the gain in ASM. It is coded in unsigned 2's complement where bit 15 to bit12 indicates the integer part and bit 12 to bit 0 is the fractional part. The maximum gain setting is 15.99975586 (24.08 dB) and the gain step is 0.0039. Default gain setting is 1.

GAIN0 Unsigned gain setting of block 0 or ASM [0:4:12].

Register Address	Register Function	Acronym
0x8008F034	ASM Gain Setting Register for Block 1	ASM_GAIN1
0x8008F054	ASM Gain Setting Register for Block 2	ASM_GAIN2

Table 95 ***Gain Setting Registers***

0x8008F018 ASM Interrupt Counter Setting Register for Block 0 ASM_IR_CNT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IR_CNT0**															
Type	R/W															
Reset	0000_0000_0000_0000															

This register sets the counter initialize value for ASM to tell MCU to update the input PCM buffer0. The counter value is decided only according to the input data rate. The default value is 0 which will not generate interrupt to MCU.

IR_CNT0 Initial counter value for interrupt generation.

Register Address	Register Function	Acronym
0x8008F038	ASM Interrupt Counter Setting Register for Block 1	ASM_IR_CNT1
0x8008F058	ASM Interrupt Counter Setting Register for Block 2	ASM_IR_CNT2

Table 96 ***Gain Setting Registers***

0x8008F108 ASM Interrupt Status Register ASM_IR_STATUS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**IR_EDI**	**IR_BLK2**	**IR_BLK1**	**IR_BLK0**
Type													RO	RO	RO	RO
Reset													0	0	0	0

This register is an indicator for MCU to see which interrupt are raised.

0x8008F10C ASM Interrupt Clear Register ASM_IR_CLR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**IR_EDI_CLR**	**IR_CLR2**	**IR_CLR1**	**IR_CLR0**
Type													WO	WO	WO	WO
Reset													-	-	-	-

This register is for MCU to clear the ASM_IR_STATUS register contents while the interrupt is read by MCU.

0x8008F110 ASM GMC Input Monitor Register ASM_GMC_READ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARCH_1X**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ALCH_1X**															
Type	RO															

This Register monitors the input of ASM from GMC.

© 2009 MediaTek Inc.

0x8008F11C ASM State0 Register ASM_STATE0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			ASM_STATE										OUT_STG_RDY			
Type			RO										RO			
Reset			0000000001										0000			

This register is for MCU to clear the ASM_IR_STATUS register contents while the interrupt is read by MCU.

0x8008F120 ASM Output Check Sum Register ASM_CHECK_SUM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ASM_CHECK_SUM															
Type	RO															
Reset	-															

This register is for MCU to check the ASM output.

0x8008F124 ASM Sine-Table Control Register ASM_SIN_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			SIN_EN	MUTE_SW		AMP_DIV			FREQ_DIV							
Type			R/W	R/W		R/W			R/W							
Reset			0	0		111			00000001							

This Register is for ASM to generate sine wave directly to AFE without other MCU Control.

SIN_EN Sent sine wave to AFE

MUTE_SW Mute sine-wave output

- **0** No mute
- **1** Mute

AMP_DIV Sine wave amplitude

- **000** 1/128x
- **001** 1/64x
- **010** 1/32x
- **011** 1/16x
- **100** 1/8x
- **101** 1/4x
- **110** 1/2x
- **111** 1x

FREQ_DIV Sine wave frequency

- **00000001** 1X
- **00000010** 2X
- **00000100** 3X
- **00001000** 4X
- **00010000** 8X
- **00100000** 16X

5.2.2.2 AFE

0x8008F200 AFE Audio MCU Control Register 0 ASM_AFE_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																AAFE ON
Type																R/W
Reset																0

MCU sets this register to start AFE audio operation. A synchronous reset signal is issued.

AAFEON AFE switch
- **0** off
- **1** on

0x8008F204 AFE Audio Control Register 1 ASM_AFE_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				MONO	ARAMPSP		ATTGAIN						AMUTEL	AMUTER	EQL_EN	EQR_EN
Type				R/W	R/W		R/W						R/W	R/W	R/W	R/W
Reset				0	00		100000						0	0	0	0

MCU sets this register for AFE operation

MONO Mono mode select. AFE HW will do (left + right) / 2 operation to the audio sample pair. Thus both DAC at right/left channels will have the same inputs.
- **0** Disable mono mode.
- **1** Enable mono mode.

ARAMPSP ramp up/down speed selection. AFS is 50.78125 kHz.
- **00** 8, 4096/AFS
- **01** 16, 2048/AFS
- **10** 24, 1024/AFS
- **11** 32, 512/AFS

AMUTER Mute the audio R-channel, with a soft ramp up/down.
- **0** No mute
- **1** Mute

AMUTEL Mute the audio L-channel, with a soft ramp up/down.
- **0** No mute
- **1** Mute

EQR_EN Enable signal for right channel equalizer.
- **0** disable
- **1** enable

EQL_EN Enable signal for left channel equalizer.
- **0** disable
- **1** enable

ATTGAIN Gain Setting for AFE input (0/32 ~ 63/32), Please set to 20.
- **111111** 63/32

100000 32/32

0x8008F208 AFE Audio Control Register 2 ASM_AFE_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RLSWT	SDM_SIGNEXT	ASDMCK_PHASE	SDMSEL	SEL_IDWA	ADITH_VAL		ADITH_ON
Type									R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset									0	1	0	0	1	00		0

MCU sets this register for SDM settings.

ADITHON Turn on the audio dither function.
- **0** off
- **1** on

ADITHVAL Dither scaling setting.
- **00** 1/4
- **01** 1/2
- **10** 1
- **11** 2

SEL_IDWA IDWA function selection.
- **0** Disable IDWA
- **1** Enable IDWA

SDMSEL Selection of SDM
- **0** original 3rd order SDM.
- **1** new 2nd order SDM.

ASDMCK_PHASE Output clock phase to analog.
- **0** Normal phase
- **1** Inverse phase

SDM_SIGNEXT Sign extension for internal data processing of 3rd order SDM.

RLSWT switch of R/L channel.
- **0** no switch
- **1** switch

0x8008F240 ~0x8008F2F0 AFE Audio Equalizer Filter Coefficient Register AFE_EQCOEF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	A															
Type	WO															

Audio front-end provides a 45-tap equalizer filter. The filter is shown below.
Note: Please make sure that ABS(A44) + ABS(A43) + … + ABS(A1) + ABS(A0) < 32768 x 16.
DO = (A44 X DI44 + A43 X DI43 … + A1 X DI1 + A0 X DI0)/32768.
DIn is the input data, and An is the coefficient of the filter, which is a 16-bit 2’s complement signed integer.
DI0 is the last input data.

The coefficient cannot be programmed when the audio path is enabled, or unpredictable noise may be generated. If coefficient programming is necessary while the audio path is enabled, the audio path must be muted during programming. After programming is complete, the audio path is not to be resumed (unmated) for 100 sampling periods.

A Coefficient of the filter.

Address	Coefficient	Address	Coefficient	Address	Coefficient
0x8008F240	A0	0x8008F27C	A15	0x8008F2B8	A30
0x8008F244	A1	0x8008F280	A16	0x8008F2BC	A31
0x8008F248	A2	0x8008F284	A17	0x8008F2C0	A32
0x8008F24C	A3	0x8008F288	A18	0x8008F2C4	A33
0x8008F250	A4	0x8008F28C	A19	0x8008F2C8	A34
0x8008F254	A5	0x8008F290	A20	0x8008F2CC	A35
0x8008F258	A6	0x8008F294	A21	0x8008F2D0	A36
0x8008F25C	A7	0x8008F298	A22	0x8008F2D4	A37
0x8008F260	A8	0x8008F29C	A23	0x8008F2D8	A38
0x8008F264	A9	0x8008F2A0	A24	0x8008F2DC	A39
0x8008F268	A10	0x8008F2A4	A25	0x8008F2E0	A40
0x8008F26C	A11	0x8008F2A8	A26	0x8008F2E4	A41
0x8008F270	A12	0x8008F2Ac	A27	0x8008F2E8	A42
0x8008F274	A13	0x8008F2B0	A28	0x8008F2EC	A43
0x8008F278	A14	0x8008F2B4	A29	0x8008F2F0	A44

0x8008F120 AFE Output Check Sum Register AFE_CHECKSUM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AFE_CHECKSUM**															
Type	RO															
Reset	-															

This register is for MCU to check the AFE output

5.2.2.3 I2S

0x8008F300 AFE EDI Control Register AFE_EDI_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**DIR**	**SRC**	**WCYCLE**					**FMT**	**EN**
Type								R/W	R/W	R/W					R/W	R/W
Reset								0	0	01111					0	0

This register is used to control the EDI

EN Enable EDI. When EDI is disabled, EDI_DAT and EDI_WS hold low.

- **0** disable EDI
- 1 enable EDI

FMT EDI format

- **0** EIAJ
- 1 I2S

WCYCLE Clock cycle count in a word. Cycle count = WCYCLE + 1, and WCYCLE can be 15 or 31 only. Any other values result in an unpredictable error.

15 Cycle count is 16.

31 Cycle count is 32.

SRC I2S clock and WS signal source.

0 Internal mode. The clock and word select signals are fed to external device from AFE.

1 External mode. The clock and word select signals are fed externally from the connected device. There is a buffer control mechanism to deal with the clock mismatch between internal and external clocks.

DIR Serial data bit direction

0 Output mode. Audio data is fed out to the external device.

1 Input mode or recording mode. By this recording mechanism, MCU can do some post processing or voice memos.

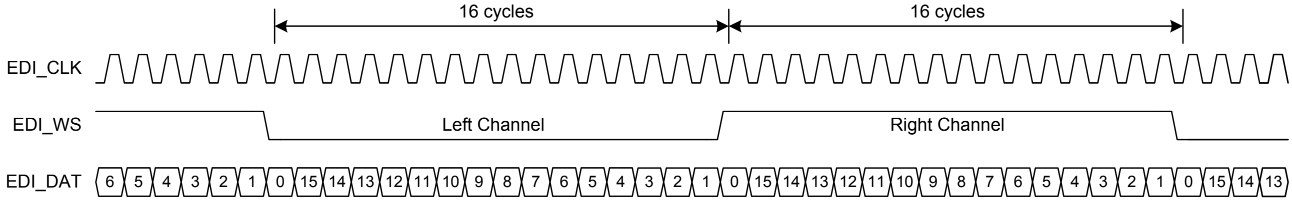


Figure 114 ***Cycle count is 16 for I2S format.***

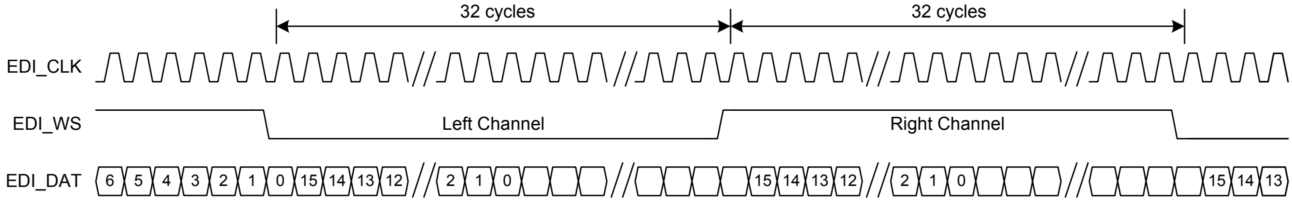


Figure 115 ***Cycle count is 32 for I2S format.***

0x8008F304 AFE EDI Output Ring Buffer Base ASM_EDI_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EDI_BASE[31:16]**															
Type	R/W															
Reset	0000_0000_0000_0000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EDI_BASE[15:0]**															
Type	R/W															
Reset	0000_0000_0000_0000															

The register indicates the starting address of the output data for ARM9. Please always set EDI_BASE[3:0] = 4'h0 for the convenience of the hardware implementation.

EDI_BASE Starting address of output EDI data ring buffers.

0x8008F308 AFE EDI Output Ring Buffer END ASM_EDI_END

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EDI_END[31:16]															
Type	R/W															
Reset	0000_0000_0000_0000															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EDI_END[15:0]															
Type	R/W															
Reset	0000_0000_0000_0000															

The register indicates the starting address of the output data for ARM9. Please always set EDI_END[3:0] = 4'hf for the convenience of the hardware implementation.

EDI_END End address of output EDI data ring buffers.

0x8008F30C AFE EDI Control Register 1 ASM_EDI_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			EDI_NUM													
Type			R/W													
Reset			00_0111_1000_0000													

This register sets the numerator for EDI enable signals. For EDI output mode, please always set EDI_NUM = 13'h0780 for 48 kHz sampling rate data. For EDI input mode, 3 settings are suggested: EDI_NUM = 13'h0500 for 32 kHz input data rate; EDI_NUM = 13'h06E4 for 44.1 kHz input data rate; and EDI_NUM = 13'h0780 for 48 kHz input data rate. Please make sure the setting is correct before EDI is opened.

EDI_NUM Numerator for EDI enable signals

0x8008F310 ASM Interrupt Counter Setting Register for EDI ASM_EDI_IR_CNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EDI_IR_CNT															
Type	R/W															
Reset	0000_0000_0000_0000															

This register sets the counter initialize value for ASM to tell MCU to get the output EDI buffer. The counter value is decided only according to the EDI output data rate. The default value is 0 which will not generate interrupt to MCU.

EDI_IR_CNT Initial counter value for interrupt generation.

0x8008F314 ASM EDI Output Monitor Register ASM_EBIT_RDATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EBIT_RDATA[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EBIT_RDATA[15:0]															
Type	RO															

This Register monitors the output of EDI to ARM9.

0x8008F318 AFE EDI Output Ring Buffer Cursor ASM_EDI_CUR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	EDI_CUR[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EDI_CUR[15:0]															
Type	RO															

The register indicates the current address of the output data for ARM9.

0x8008F31C ASM EDI Input Monitor Register ASM_EDI_DATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EDI_DATA[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EDI_DATA[15:0]															
Type	RO															

This Register monitors the output of EDI to ARM9.

5.2.3 Programming Guide

1. The input buffer length of the MCU must longer than the desired length for the hardware margin.
2. Power On ASM ,AFE and GMC1 clock by setting GRAPH1SYS_CG_CLR (0x80092340) = 0x00120001.
3. ASM will start whenever the ASMON in ASM_CON0 comes, please set all registers before the ASMON is raised
4. The settings in ASM_SET and ASM_GAIN of each block can't be changed until an audio wave is finished.
5. The operation sequence for ASM and AFE (I2S) is to start AFE (I2S) before ASM. While the operation is finished, stop AFE (I2S) before stopping ASM.
6. ASM stopping sequence
   a、 Close all ASM blocks (ASM_CON0 = 0x0000).
   b、 Clear all IRQ status (ASM_IR_STATUS).
   c、 Power down ASM clock.
7. AFE and I2S output can't be opened at the same time.
8. I2S input mode :
   a、 Power on ASM by setting GRAPH1SYS_CG_CLR (0x80092340) = 0x00100000.
   b、 Set ASM_EDI_BASE, ASM_EDI_END, ASM_EDI_CON1 and ASM_EDI_IR_CNT to the desired settings.
   c、 Set DIR, SRC and EN of AFE_EDI_CON to 1 to Start I2S input mode operation.
9. Once ASM is closed, please wait at least 100us to reopen it.

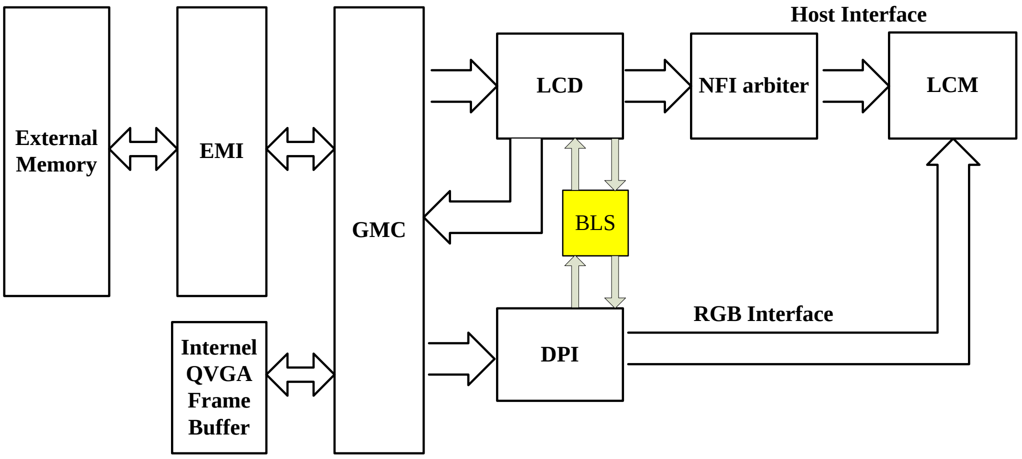
5.3 Backlight Scaling

5.3.1 Introduction

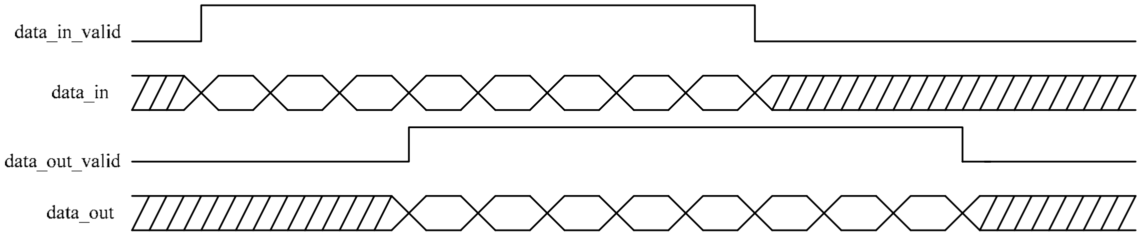
Back light scaling is used to modify content and back light to preserve original luminance and save the power. We provide the path for LCD and DPI.

.

5.3.2 Architecture of BLS

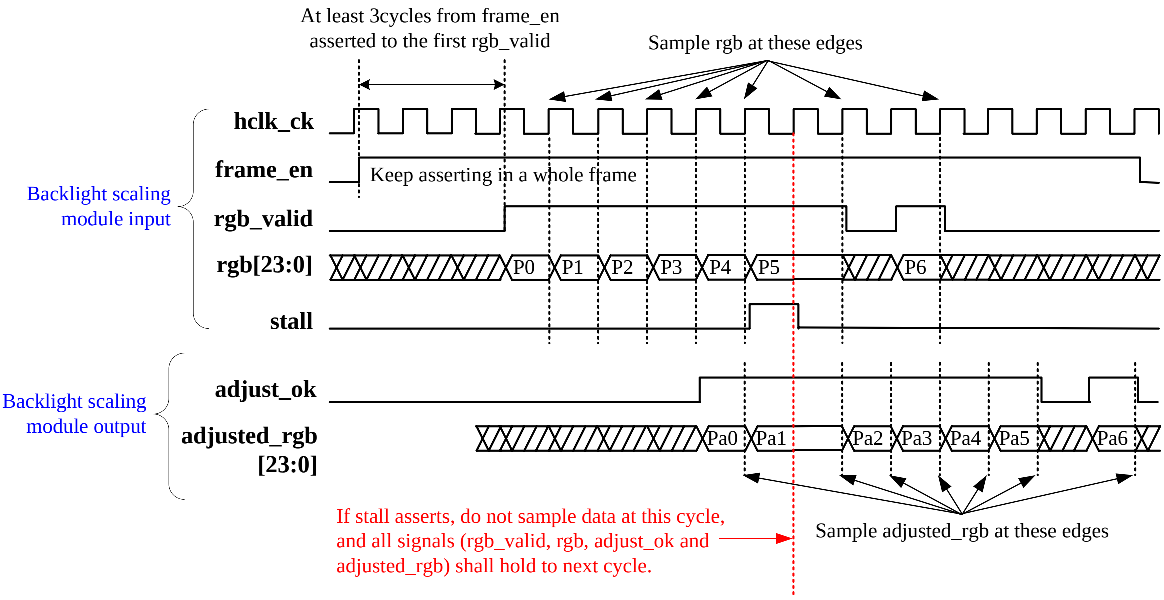


DPI Interface:



LCD Interface:

 © 2009 MediaTek Inc.



5.3.3 Register Definitions

1. Table0

BLS + 0000h — **BLS_PWM_MAX_CONTENTA**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_MAX_CONTENT00[7:0]								PWM_MAX_CONTENT01[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_MAX_CONTENT02[7:0]								PWM_MAX_CONTENT03[7:0]							
Type	R/W								R/W							
Reset	0								0							

2. Table0

BLS + 0004h — **BLS_PWM_MAX_CONTENTB**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_MAX_CONTENT04[7:0]								PWM_MAX_CONTENT05[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_MAX_CONTENT06[7:0]								PWM_MAX_CONTENT07[7:0]							
Type	R/W								R/W							
Reset	0								0							

3. Table0

BLS + 0008h BLS_PWM_MAX_CONTENTC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_MAX_CONTENT08[7:0]								PWM_MAX_CONTENT09[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_MAX_CONTENT10[7:0]								PWM_MAX_CONTENT11[7:0]							
Type	R/W								R/W							
Reset	0								0							

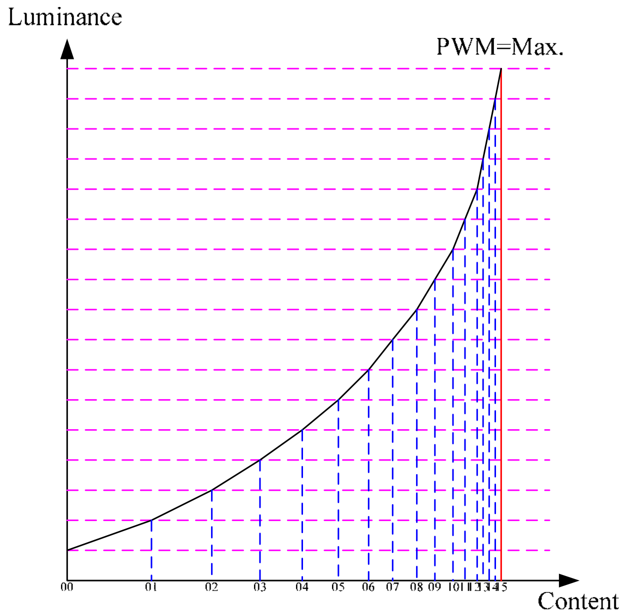
4. Table0

BLS + 000Ch BLS_PWM_MAX_CONTENTD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_MAX_CONTENT12[7:0]								PWM_MAX_CONTENT13[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_MAX_CONTENT14[7:0]								PWM_MAX_CONTENT15[7:0]							
Type	R/W								R/W							
Reset	0								0							

Comments:

We divide the curve of content vs. luminance with PWM=maximum at most 16 pieces and records the content of each piece. The software can decide the level num of the table.



Level=16
PWM=Max.
255
content15
content14
content13
content12
content11
content10
content09
content08
content07
content06
content05
content04
content03
content02
content01
content00=0

Level=8
PWM=Max.
255
content07
content06
content05
content04
content03
content02
content01
content00=0

Level=4
PWM=Max.
255
content03
content02
content01
content00=0

5. Table1

BLS + 0010h — BLS_PWM_APPLIED_CONTENTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_ APPLIED _CONTENT00[7:0]								PWM_ APPLIED _CONTENT01[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_ APPLIED _CONTENT02[7:0]								PWM_ APPLIED _CONTENT03[7:0]							
Type	R/W								R/W							
Reset	0								0							

6. Table1

BLS + 0014h — BLS_PWM_ APPLIED _CONTENTB

Bit	31	30	29	28	27	26	25	24	23		21	20	19	18	17	16
Name	PWM_ APPLIED _CONTENT04[7:0]								PWM_ APPLIED _CONTENT05[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7		5	4	3	2	1	0
Name	PWM_ APPLIED _CONTENT06[7:0]								PWM_ APPLIED _CONTENT07[7:0]							
Type	R/W								R/W							
Reset	0								0							

7. Table1

BLS + 0018h — BLS_PWM_ APPLIED _CONTENTC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM APPLIED _CONTENT08[7:0]								PWM_ APPLIED _CONTENT09[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_ APPLIED _CONTENT10[7:0]								PWM_ APPLIED _CONTENT11[7:0]							
Type	R/W								R/W							
Reset	0								0							

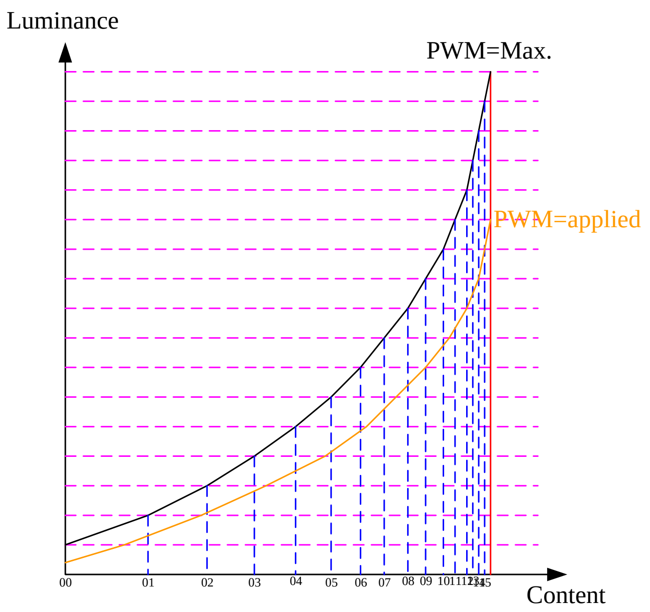
8. Table1

BLS + 001Ch — BLS_PWM_ APPLIED _CONTENTD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_ APPLIED _CONTENT12[7:0]								PWM_ APPLIED _CONTENT13[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM_ APPLIED _CONTENT14[7:0]								PWM_ APPLIED _CONTENT15[7:0]							
Type	R/W								R/W							
Reset	0								0							

Comments:

We divide the curve of content vs. luminance with PWM=applied at most 16 pieces and records the content of each piece. The untouched region when content reached its max (255) all fill with 255.



Level=16
PWM=Applied
255
content15=255
content14=255
content13=255
content12=255
content11=255
content10
content09
content08
content07
content06
content05
content04
content03
content02
content01
content00

Level=8
PWM=Applied
255
content7=255
content6=255
content05
content04
content03
content02
content01
content00

Level=4
PWM=Applied
255
content3=255
content02
content01
content00

9. Table2

BLS + 0020h BLS_APPLIED_CONTRASTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST00[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST01[15:0]															
Type	R/W															
Reset	0															

10. Table2

BLS + 0024h BLS_APPLIED_CONTRASTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST02[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST03[15:0]															
Type	R/W															
Reset	0															

11. Table2

BLS + 0028h BLS_APPLIED_CONTRASTC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST04[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST05[15:0]															
Type	R/W															
Reset	0															

12. Table2

BLS + 002Ch BLS_APPLIED_CONTRASTD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST06[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST07[15:0]															
Type	R/W															
Reset	0															

13. Table2

BLS + 0030h BLS_APPLIED_CONTRASTE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST08[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST09[15:0]															
Type	R/W															
Reset	0															

14. Table2

BLS + 0034h BLS_APPLIED_CONTRASTF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST10[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST11[15:0]															
Type	R/W															
Reset	0															

15. Table2

BLS + 0038h BLS_APPLIED_CONTRASTG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST12[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST13[15:0]															
Type	R/W															
Reset	0															

16. Table2

BLS + 003Ch BLS_APPLIED_CONTRASTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLS_APPLIED_CONTRAST14[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLS_APPLIED_CONTRAST15[15:0]															
Type	R/W															
Reset	0															

Comments:

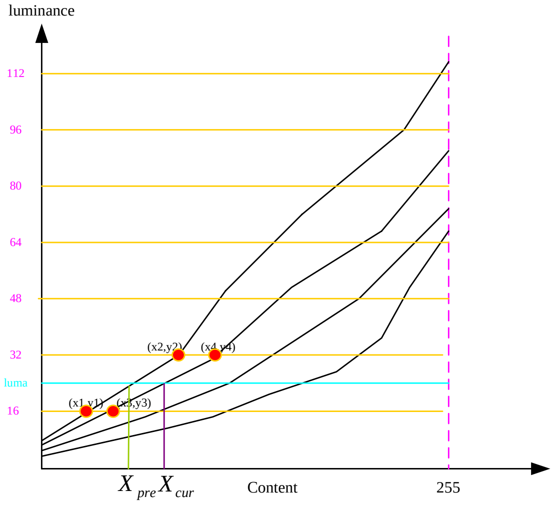
We calculate the contrast using the Table0 and Table1 according to the following formula.

$$X_{cur} = \frac{(x_4 - x_3)}{(x_2 - x_1)}(X_{pre} - x_1) + x_3$$

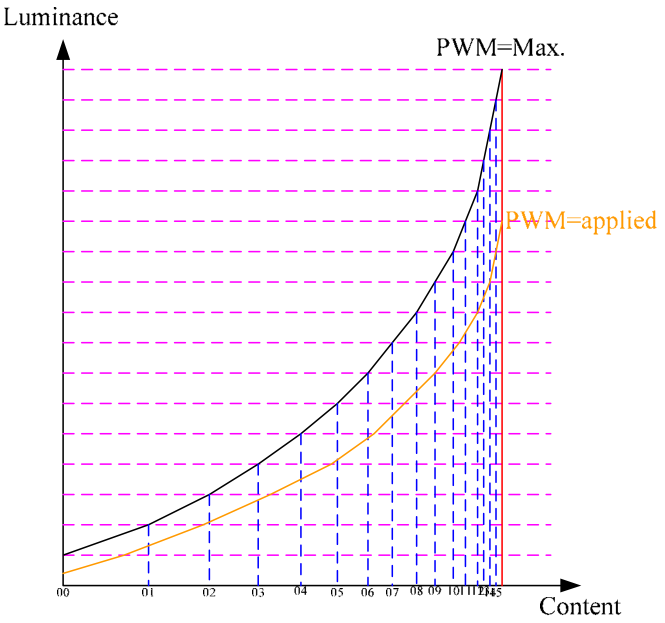
$$Contrast = \frac{(x_4 - x_3)}{(x_2 - x_1)},$$

where x_4, x_3 means table1 (applied PWM)'s content

where x_2, x_1 means table0 (Max. PWM)'s content



© 2009 MediaTek Inc.



PWM=Applied	PWM=Max.
255	255
content15=255	content15
content14=255	content14
content13=255	content13
content12=255	content12
content11=255	content11
content10	content10
content09	content09
content08	content08
content07	content07
content06	content06
content05	content05
content04	content04
content03	content03
content02	content02
content01	content01
content00	content00=0

Table2
0
contrast15=0
contrast14=0
contrast13=0
contrast12=0
contrast11=0
contrast10
contrast09
contrast08
contrast07
contrast06
contrast05
contrast04
contrast03
contrast02
contrast01
contrast00

17. Machine1 Contrast

BLS + 0040h **BLS_MACHINE1_CONTRAST**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M1_CONTRAST[15:0]															
Type	R/W															
Reset	0															

Comment:

M1_CONTRAST: 16 bits unsigned floating point, 8 bits unsigned integer 8 bits unsigned decimal.

18. Machine1 Brightness

BLS + 0044h **BLS_MACHINE1_BRIGHTNESS**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								M1_BRIGHTNESS1[8:0]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								M1_BRIGHTNESS0[8:0]								
Type								R/W								
Reset								0								

Comment:

M1_BRIGHTNESS1: 9 bits signed integer with 2's complement.

M1_BRIGHTNESS0: 9 bits signed integer with 2's complement.

19. Machine1 Selector

BLS + 0048h **BLS_MACHINE1_SELECTOR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											M1_MODE_SEL[2:0]			M1_SRC_SEL[2:0]		
Type											R/W			R/W		
Reset											0			0		

Comment:

M1_MODE_SEL[0]: 1->rgb_in_content, 0->sat1 output
M1_MODE_SEL[1]: 1->sat3 output, 0->sat4 output
M1_MODE_SEL[2]: 1->mux output, 0->rgb_in_content
M1_SRC_SEL[0] : 1->table output content(-), 0->m1_brightness0
M1_SRC_SEL[1] : 1->table output contrast, 0->m1_contrast
M1_SRC_SEL[2] : 1->table output content(+), 0->m1_brightness1

20. Machine2 Contrast

BLS + 004Ch **BLS_MACHINE2_CONTRAST**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M2_CONTRAST[15:0]															
Type	R/W															
Reset	0															

Comment:

M2_CONTRAST: 16 bits unsigned floating point, 8 bits unsigned integer 8 bits unsigned decimal.

21. Machine2 Brightness

BLS + 0050h **BLS_MACHINE2_BRIGHTNESS**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								M2_BRIGHTNESS1[8:0]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								M2_BRIGHTNESS0[8:0]								
Type								R/W								
Reset								0								

Comment:

M2_BRIGHTNESS1: 9 bits signed integer with 2's complement.
M2_BRIGHTNESS0: 9 bits signed integer with 2's complement.

22. Machine2 Selector

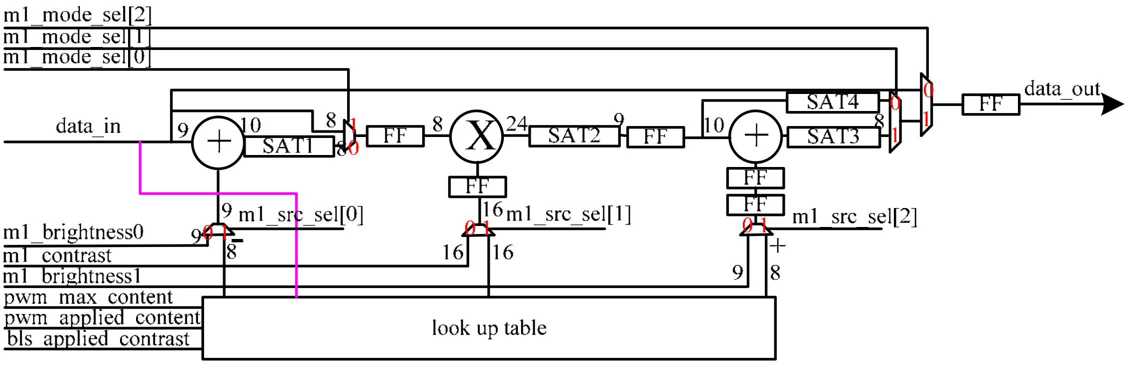
BLS + 0054h BLS_MACHINE2_SELECTOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**M2_MODE_SEL[2:0]**			**M2_SRC_SEL[2:0]**		
Type											**R/W**			**R/W**		
Reset											**0**			**0**		

Comment:

M2_MODE_SEL[0]: 1->rgb_in_content, 0->sat1 output
M2_MODE_SEL[1]: 1->sat3 output, 0->sat4 output
M2_MODE_SEL[2]: 1->mux output, 0->rgb_in_content
M2_SRC_SEL[0] : 1->table output content(-), 0->m2_brightness0
M2_SRC_SEL[1] : 1->table output contrast, 0->m2_contrast
M2_SRC_SEL[2] : 1->table output content(+), 0->m2_brightness1

Comment:

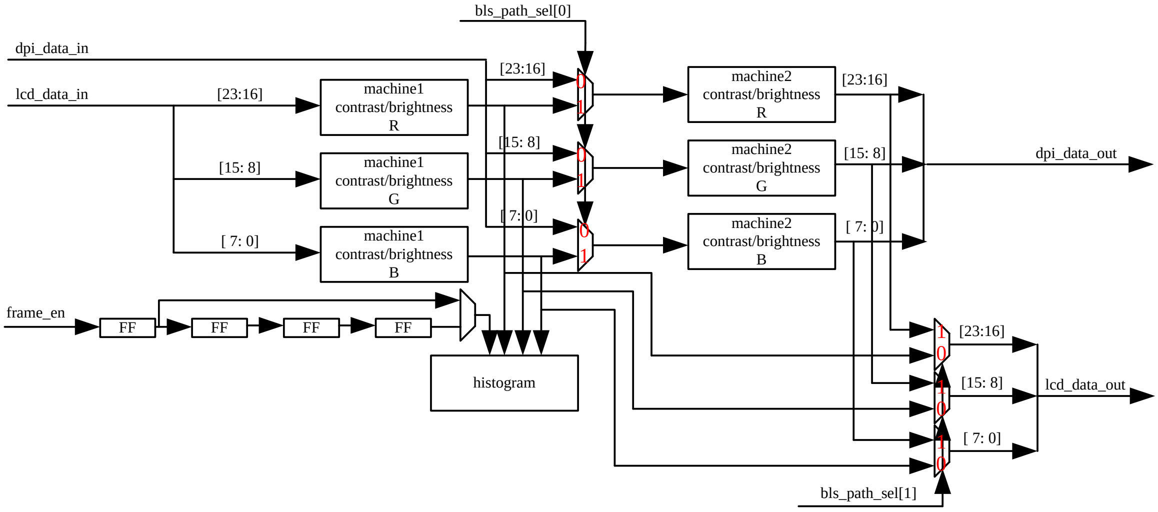


23. Path Selector

BLS + 0058h BLS_SETTING

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**BLS_LEVEL_NUM[3:0]**			
Type													**R/W**			
Reset													**0**			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**BLS_PATH_SEL[1:0]**	
Type															**R/W**	
Reset															**0**	

Comment:



BLS_LEVEL_NUM: 1~16 levels. 0 means 1 level, 15 means 16 levels
BLS_PATH_SEL[0]: 0->DPI, 1->machine1
BLS_PATH_SEL[1]: 0->machine1, 1-> machine2

24. Histogram BIN00

BLS + 005Ch **BLS_HISTOGRAM BIN00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN00[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN00[15:0]															
Type	R															
Reset	0															

25. Histogram BIN01

BLS + 0060h **BLS_HISTOGRAM BIN01**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN01[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN01[15:0]															
Type	R															
Reset	0															

26. Histogram BIN02

BLS + 0064h **BLS_HISTOGRAM BIN02**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN02[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN02[15:0]**															
Type	**R**															
Reset	**0**															

27. Histogram BIN03

BLS + 0068h **BLS_HISTOGRAM BIN03**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN03[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN03[15:0]**															
Type	**R**															
Reset	**0**															

28. Histogram BIN04

BLS + 006Ch **BLS_HISTOGRAM BIN04**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN04[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN04[15:0]**															
Type	**R**															
Reset	**0**															

29. Histogram BIN05

BLS + 0070h **BLS_HISTOGRAM BIN05**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN05[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN05[15:0]**															
Type	**R**															
Reset	**0**															

30. Histogram BIN06

BLS + 0074h **BLS_HISTOGRAM BIN06**

© 2009 MediaTek Inc.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN06[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN06[15:0]															
Type	R															
Reset	0															

31. Histogram BIN07

BLS + 0078h BLS_HISTOGRAM BIN07

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN07[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN07[15:0]															
Type	R															
Reset	0															

32. Histogram BIN08

BLS + 007Ch BLS_HISTOGRAM BIN08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN08[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN08[15:0]															
Type	R															
Reset	0															

33. Histogram BIN09

BLS + 0080h BLS_HISTOGRAM BIN09

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN09[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN09[15:0]															
Type	R															
Reset	0															

34. Histogram BIN00

BLS + 0084h BLS_HISTOGRAM BIN10

© 2009 MediaTek Inc.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN10[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN10[15:0]															
Type	R															
Reset	0															

35. Histogram BIN11

BLS + 0088h BLS_HISTOGRAM BIN11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN11[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN11[15:0]															
Type	R															
Reset	0															

36. Histogram BIN12

BLS + 008Ch BLS_HISTOGRAM BIN12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN12[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN12[15:0]															
Type	R															
Reset	0															

37. Histogram BIN13

BLS + 0090h BLS_HISTOGRAM BIN13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												BIN13[20:16]				
Type												R				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN13[15:0]															
Type	R															
Reset	0															

38. Histogram BIN14

BLS + 0094h BLS_HISTOGRAM BIN14

© 2009 MediaTek Inc.

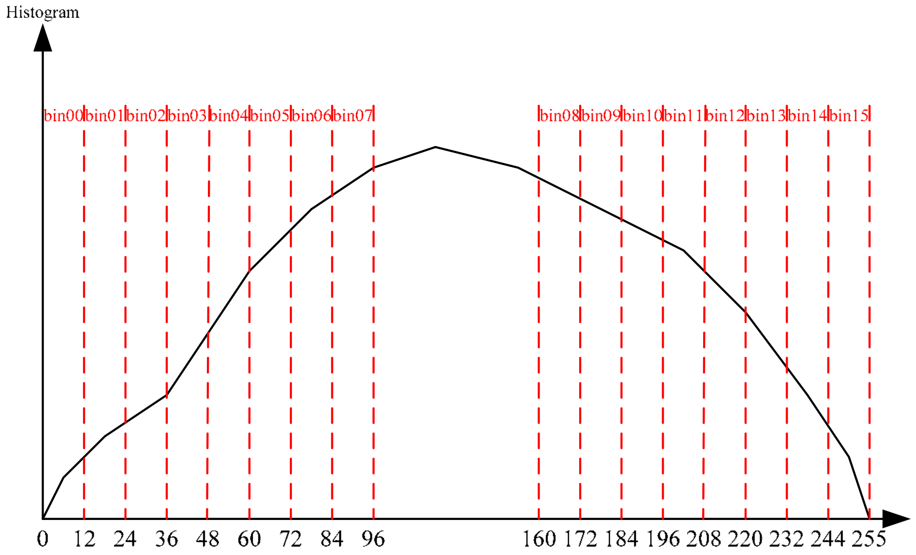
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN14[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN14[15:0]**															
Type	**R**															
Reset	**0**															

39. Histogram BIN15

BLS + 0098h — **BLS_HISTOGRAM BIN15**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**BIN15[20:16]**				
Type												**R**				
Reset												**0**				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIN15[15:0]**															
Type	**R**															
Reset	**0**															

Comment:



40. Histogram Enable

BLS + 009Ch — **BLS_HISTOGRAM ENABLE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**HIS_EN**

Type																R/W
Reset																0

Comment:

EN=1: Histogram start to active

EN=0: Histogram not work

41. Histogram Clear

BLS + 00A0h BLS_HISTOGRAM_CLEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																HIS_CLR
Type																R/W
Reset																0

Comment:

CLR=1: Clear the bin00~bin15 to 0, should be turn on when en=0

CLR=0: No clear

42. Histogram Done

BLS + 00A4h BLS_HISTOGRAM_DONE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																HIS_DONE
Type																R/W
Reset																0

Comment:

DONE=1: Histogram has finished one frame

DONE=0: Histogram hasn't finished one frame

The done should be clear when software finish his job.

43. Histogram Done Mask

BLS + 00A8h BLS_HISTOGRAM_DONE_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																HIS_DONE_EN
Type																R/W
Reset																0

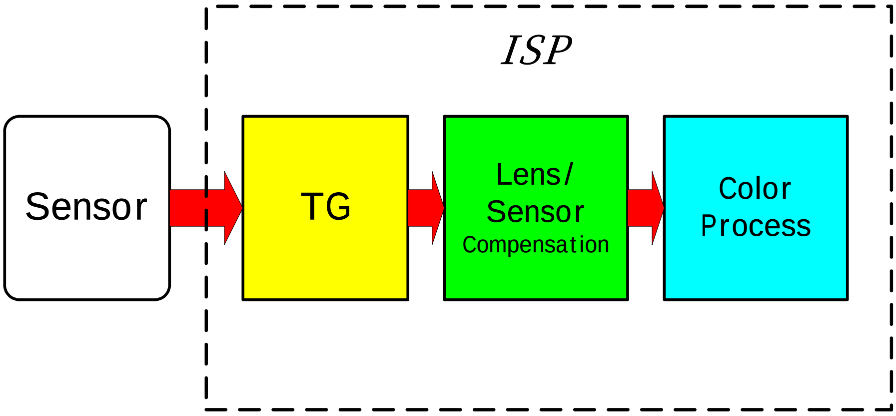
Comment:
Histogram's interrupt mask it will be active when his_done = 1 and his_done_en = 1
44. Histogram Setting

BLS + 00ACh **BLS_HISTOGRAM_SETTING**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															HIS_DONEC	HIS_ACLR
															R/W	R/W
															0	0

Comment:
DONEC: Done auto continue. When histogram finish one frame and turn on this bit it will auto continue to calculate next frame's histogram.
ACLR: Auto Clear. When histogram finish one frame and turn on this bit it will auto clear the bin data when next frame is start.

5.4 Camera Interface



MT6516 ISP incorporates a feature rich image signal processor to connect with a variety of image sensor components. This processor consists of timing generated unit (TG) and lens/sensor compensation unit and image process unit.

Timing generated unit (TG) cooperates with master type image sensor only. That means sensor should send vertical and horizontal signals to TG. TG offers sensor required data clock and receive sensor Bayer pattern raw data by internal auto synchronization or external pixel clock synchronization. The main purpose of TG is to create data clock for master type image sensor and accept vertical/horizontal synchronization signal and sensor data, and then generate grabbed area of bayer data or YUV422/RGB565 data to the lens/sensor compensation unit.

Lens/sensor compensation unit generates compensated raw data to the color process unit in Bayer raw data input mode. In YUV422/RGB565 input mode, this stage is bypassed.

Color process unit accepts Bayer pattern raw data or YUV422/RGB565 data that is generated by lens/sensor compensation unit. The output of ISP is YCbCr 888 data format which can be easily encoded by the compress engine (JPEG encoder and MPEG4 encoder). It can be the basic data domain of other data format translation such as R/G/B domain. The ISP is pipelined, and during processing stages ISP hardware can auto extract meaningful information for further AE/AF/AWB calculation. These information are temporary stored on ISP registers or memory and can be read back by MCU.

5.4.1 Register Table

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CAM + 0000h	TG Phase Counter Register	CAM_PHSCNT
CAM + 0004h	Sensor Size Configuration Register	CAM_CAMWIN
CAM + 0008h	TG Grab Range Start/End Pixel Configuration Register	CAM_GRABCOL
CAM + 000Ch	TG Grab Range Start/End Line Configuration Register	CAM_GRABROW
CAM + 0010h	Sensor Mode Configuration Register	CAM_CSMODE
CAM + 0014h	Component R, Gr, B, Gb, Offset Adjustment Register	CAM_RGBOFF
CAM + 0018h	View Finder Mode Control Register	CAM_VFCON
CAM + 001Ch	Camera Module Interrupt Enable Register	CAM_INTEN
CAM + 0020h	Camera Module Interrupt Status Register	CAM_INTSTA
CAM + 0024h	Camera Module Path Config Register	CAM_PATH
CAM + 0028h	Camera Module Input Address Register	CAM_INADDR
CAM + 002Ch	Camera Module Output Address Register	CAM_OUTADDR
CAM + 0030h	Preprocessing Control Register 1	CAM_CTRL1
CAM + 0034h	AWB R,G, B Gain Control Register 1	CAM_RGBGAIN1
CAM + 0038h	AWB R,G, B Gain Control Register 2	CAM_RGBGAIN2
CAM + 003Ch	Histogram Boundary Control Register 1	CAM_HIS0
CAM + 0040h	Histogram Boundary Control Register 2	CAM_HIS1
CAM + 0044h	Preprocessing Control Register 2	CAM_CTRL2
CAM + 0048h	AE Window 1 Register	CAM_AEWIN1
CAM + 004Ch	AE Histogram Window Register	CAM_ AEWIN2
CAM + 0050h	AE Histogram Gain Register	CAM_ AEWIN3
CAM + 0054h	Global Shutter Control Register	CAM_GSCTRL
CAM + 0058h	Mechanical Shutter Control Register	CAM_MSCTRL
CAM + 005Ch	Mechanical Shutter M1 Setting Register	CAM_MS1TIME
CAM + 0060h	Mechanical Shutter M2 Setting Register	CAM_MS2TIME
CAM + 0064h	Reserved	Reserved
CAM + 0068h	Reserved	Reserved
CAM + 006Ch	AWB Window Register	CAM_AWBWIN
CAM + 0070h	Color Processing Stage Control Register	CAM_CPSCON1
CAM + 0074h	Interpolation Register 1	CAM_INTER1

CAM + 0078h	Interpolation Register 2	CAM_INTER2
CAM + 007Ch	Edge Core Register	CAM_EDGCORE
CAM + 0080h	Edge Gain Register 1	CAM_EDGGAIN1
CAM + 0084h	Edge Gain Register 2	CAM_EDGGAIN2
CAM + 0088h	Edge Threshold Register	CAM_EDGTHRE
CAM + 008Ch	Edge Vertical Control Register	CAM_EDGVCON
CAM + 0090h	Reserved	Reserved
CAM + 0094h	Reserved	Reserved
CAM + 0098h	Reserved	Reserved
CAM + 009Ch	Color Matrix 1 Register	CAM_MATRIX1
CAM + 00A0h	Color Matrix 2 Register	CAM_MATRIX2
CAM + 00A4h	Color Matrix 3 Register	CAM_MATRIX3
CAM + 00A8h	Reserved	Reserved
CAM + 00ACh	Color Process Stage Control Register 2	CAM_CPSCON2
CAM + 00B0h	Flare Gain Register	CAM_FLREGAIN
CAM + 00B4h	Flare Offset Register	CAM_FLREOFF
CAM + 00B8h	Y Channel Configuration Register	CAM_YCHAN
CAM + 00BCh	RGB2YCC Control Register	RGB2YCC_CON
CAM + 00C0h	Reserved	Reserved
CAM + 00C4h	Reserved	Reserved
CAM + 00C8h	Reserved	Reserved
CAM + 00CCh	Reserved	Reserved
CAM + 00D0h	Reserved	Reserved
CAM + 00D4h	Reserved	Reserved
CAM + 00D8h	Reserved	Reserved
CAM + 00DCh	Reserved	Reserved
CAM + 00E0h	Reserved	Reserved
CAM + 00E4h	MIPI CSI2 Status Register 1	CAM_CSI2STA1
CAM + 00E8h	MIPI CSI2 Status Register 2	CAM_CSI2STA2
CAM + 00ECh	Reserved	Reserved
CAM + 00F0h	Reserved	Reserved
CAM + 00F4h	Reserved	Reserved
CAM + 00F8h	Reserved	Reserved
CAM + 00FCh	Reserved	Reserved
CAM + 0100h	Reserved	Reserved
CAM + 0104h	Reserved	Reserved
CAM + 0108h	Flare Histogram 1 Result	CAM_HIS_RLT0
CAM + 010Ch	Flare Histogram 2 Result	CAM_HIS_RLT1

© 2009 MediaTek Inc.

CAM + 0110h	Flare Histogram 3 Result	CAM_HIS_RLT2
CAM + 0114h	Flare Histogram 4 Result	CAM_HIS_RLT3
CAM + 0118h	Flare Histogram 5 Result	CAM_HIS_RLT4
CAM + 011Ch	Reserved	Reserved
CAM + 0120h	Reserved	Reserved
CAM + 0124h	Reserved	Reserved
CAM + 0128h	Vertical Subsample Control Register	CAM_VSUB
CAM + 012Ch	Horizontal Subsample Control Register	CAM_HSUB
CAM + 0130h	Reserved	Reserved
CAM + 0134h	Reserved	Reserved
CAM + 0138h	Reserved	Reserved
CAM + 013Ch	Reserved	Reserved
CAM + 0140h	Reserved	Reserved
CAM + 0144h	Reserved	Reserved
CAM + 0148h	Reserved	Reserved
CAM + 014Ch	Reserved	Reserved
CAM + 0150h	Reserved	Reserved
CAM + 0154h	Defect Pixel Configuration Register	CAM_DEFECT0
CAM + 0158h	Defect Pixel Table Address Register	CAM_DEFECT1
CAM + 015Ch	Defect Pixel Table Debug Register	CAM_DEFECT2
CAM + 0160h	Reserved	Reserved
CAM + 0164h	Reserved	Reserved
CAM + 0168h	Reserved	Reserved
CAM + 016Ch	Raw Gain Register 1	CAM_RAWGAIN0
CAM + 0170h	Raw Gain Register 2	CAM_RAWGAIN1
CAM + 0174h	Result Window Vertical Size Register	RWINV_SEL
CAM + 0178h	Result Window Horizontal Size Register	RWINH_SEL
CAM + 017Ch	Reserved	Reserved
CAM + 0180h	Camera Interface Debug Mode Control Register	CAM_DEBUG
CAM + 0184h	Camera Module Debug Information Write Out Destination Address	CAM_DSTADDR
CAM + 0188h	Camera Module Debug Information Last Transfer Destination Address	CAM_LSTADDR
CAM + 018Ch	Camera Module Frame Buffer Transfer Out Count Register	CAM_XFERCNT
CAM + 0190h	Sensor Test Module Configuration Register 1	CAM_MDLCFG1
CAM + 0194h	Sensor Test Module Configuration Register 2	CAM_MDLCFG2
CAM + 0198h	Reserved	Reserved
CAM + 019Ch	Reserved	Reserved
CAM + 01A0h	Reserved	Reserved

CAM + 01A4h	Reserved	Reserved
CAM + 01A8h	Gamma Register 1	GMA_REG1
CAM + 01ACh	Gamma Register 2	GMA_REG2
CAM + 01B0h	Gamma Register 3	GMA_REG3
CAM + 01B4h	Gamma Register 4	GMA_REG4
CAM + 01B8h	Gamma Register 5	GMA_REG5
CAM + 01BCh	Raw Data Accumulation Config Register	RAWACC
CAM + 01C0h	Raw Data Accumulation Window	RAWWIN
CAM + 01C4h	Raw Data Accumulation Result 1	RAWSUM0
CAM + 01C8h	Raw Data Accumulation Result 2	RAWSUM1
CAM + 01CCh	Raw Data Accumulation Result 3	RAWSUM2
CAM + 01D0h	Raw Data Accumulation Result 4	RAWSUM3
CAM + 01D4h	Reserved	Reserved
CAM + 01D8h	Cam Reset Register	CAM_RESET
CAM + 01DCh	TG Status Register	TG_STATUS
CAM + 01E0h	Flash Control 1 Register	FLASH_CTRL0
CAM + 01E4h	Flash Control 2 Register	FLASH_CTRL1
CAM + 01E8h	FlashB Control 1 Register	FLASHB_CTRL0
CAM + 01ECh	FlashB Control 2 Register	FLASHB_CTRL1
CAM + 01F0h	Flare Histogram 6 Result	CAM_HIS_RLT5
CAM + 01F4h	Flare Histogram 7 Result	CAM_HIS_RLT6
CAM + 01F8h	Flare Histogram 8 Result	CAM_HIS_RLT7
CAM + 01FCh	Flare Histogram 9 Result	CAM_HIS_RLT8
CAM + 0200h	Flare Histogram 10 Result	CAM_HIS_RLT9
CAM + 0204h	Reserved	Reserved
CAM + 0208h	Reserved	Reserved
CAM + 020Ch	Reserved	Reserved
CAM + 0210h	Reserved	Reserved
CAM + 0214h	Shading Control 1 Register	CAM_SHADING1
CAM + 0218h	Shading Control 2 Register	CAM_SHADING2
CAM + 021Ch	Shading Read Address Register	SD_RADDR
CAM + 0220h	Shading Last Block Config Register	SD_LBLOCK
CAM + 0224h	Shading Ratio Config Register	SD_RATIO
CAM + 0228h	Reserved	Reserved
CAM + 022Ch	Reserved	Reserved
CAM + 0230h	EE Control Register	EE_CTRL
CAM + 0234h	ED LUT X Configuration	ED_LUT_X
CAM + 0238h	ED LUT Y Configuration	ED_LUT_Y

© 2009 MediaTek Inc.

CAM + 023Ch	Reserved	Reserved
CAM + 0240h	Reserved	Reserved
CAM + 0244h	Reserved	Reserved
CAM + 0248h	Reserved	Reserved
CAM + 024Ch	AF Window 1 Register	CAM_AFWIN0
CAM + 0250h	AF Window 2 Register	CAM_AFWIN1
CAM + 0254h	AF Window 3 Register	CAM_AFWIN2
CAM + 0258h	AF Window 4 Register	CAM_AFWIN3
CAM + 025Ch	AF Window 5 Register	CAM_AFWIN4
CAM + 0260h	AF Threshold 1 Register	CAM_AFTH0
CAM + 0264h	AF Threshold 2 Register	CAM_AFTH1
CAM + 0268h	AF Window 6 Register	CAM_AFWIN5
CAM + 026Ch	AF Window 7 Register	CAM_AFWIN6
CAM + 0270h	AF Window 8 Register	CAM_AFWIN7
CAM + 0274h	Cam Version Register	CAM_VERSION
CAM + 027Ch	AWB Sum Window Config Register	AWBSUM_WIN
CAM + 0280h	AWB Control Register	AWB_CTRL
CAM + 0284h	AWB Threshold Config Register	AWB_TH
CAM + 0288h	AWB Color Space H1 Config Register	AWBXY_H1
CAM + 028Ch	AWB Color Space H2 Config Register	AWBXY_H2
CAM + 0290h	AWB Color Edge Window Horizontal Config Register	AWBCE_WINH
CAM + 0294h	AWB Color Edge Window Vertical Config Register	AWBCE_WINV
CAM + 0298h	AWB XY Window 1 Horizontal Config Register	AWBXY_WINH0
CAM + 029Ch	AWB XY Window 1 Vertical Config Register	AWBXY_WINV0
CAM + 02A0h	AWB XY Window 2 Horizontal Config Register	AWBXY_WINH1
CAM + 02A4h	AWB XY Window 2 Vertical Config Register	AWBXY_WINV1
CAM + 02A8h	AWB XY Window 3 Horizontal Config Register	AWBXY_WINH2
CAM + 02ACh	AWB XY Window 3 Vertical Config Register	AWBXY_WINV2
CAM + 02B0h	AWB XY Window 4 Horizontal Config Register	AWBXY_WINH3
CAM + 02B4h	AWB XY Window 4 Vertical Config Register	AWBXY_WINV3
CAM + 02B8h	AWB XY Window 5 Horizontal Config Register	AWBXY_WINH4
CAM + 02BCh	AWB XY Window 5 Vertical Config Register	AWBXY_WINV4
CAM + 02C0h	AWB XY Window 6 Horizontal Config Register	AWBXY_WINH5
CAM + 02C4h	AWB XY Window 6 Vertical Config Register	AWBXY_WINV5
CAM + 02C8h	AWB XY Window 7 Horizontal Config Register	AWBXY_WINH6
CAM + 02CCh	AWB XY Window 7 Vertical Config Register	AWBXY_WINV6
CAM + 02D0h	AWB XY Window 8 Horizontal Config Register	AWBXY_WINH7
CAM + 02D4h	AWB XY Window 8 Vertical Config Register	AWBXY_WINV7

© 2009 MediaTek Inc.

CAM + 02D8h	AWB XY Window 9 Horizontal Config Register	AWBXY_WINH8
CAM + 02DCh	AWB XY Window 9 Vertical Config Register	AWBXY_WINV8
CAM + 02E0h	AWB XY Window 10 Horizontal Config Register	AWBXY_WINH9
CAM + 02E4h	AWB XY Window 10 Vertical Config Register	AWBXY_WINV9
CAM + 02E8h	AWB XY Window 11 Horizontal Config Register	AWBXY_WINHA
CAM + 02ECh	AWB XY Window 11 Vertical Config Register	AWBXY_WINVA
CAM + 02F0h	AWB XY Window 12 Horizontal Config Register	AWBXY_WINHB
CAM + 02F4h	AWB XY Window 12 Vertical Config Register	AWBXY_WINVB
CAM + 02F8h	AWB Sum Window Paxel Count	AWBSUM_PCNT
CAM + 02FCh	AWB Sum Window R Sum	AWBSUM_RSUM
CAM + 0300h	AWB Sum Window G Sum	AWBSUM_GSUM
CAM + 0304h	AWB Sum Window B Sum	AWBSUM_BSUM
CAM + 0308h	AWB Color Edge Window Paxel Count	AWBCE_PCNT
CAM + 030Ch	AWB Color Edge Window R Sum	AWBCE_RSUM
CAM + 0310h	AWB Color Edge Window G Sum	AWBCE_GSUM
CAM + 0314h	AWB Color Edge Window B Sum	AWBCE_BSUM
CAM + 0318h	AWB XY Window 1 Paxel Count	AWBXY_PCNT0
CAM + 031Ch	AWB XY Window 1 R Sum	AWBXY_RSUM0
CAM + 0320h	AWB XY Window 1 G Sum	AWBXY_GSUM0
CAM + 0324h	AWB XY Window 1 B Sum	AWBXY_BSUM0
CAM + 0328h	AWB XY Window 2 Paxel Count	AWBXY_PCNT1
CAM + 032Ch	AWB XY Window 2 R Sum	AWBXY_RSUM1
CAM + 0330h	AWB XY Window 2 G Sum	AWBXY_GSUM1
CAM + 0334h	AWB XY Window 2 B Sum	AWBXY_BSUM1
CAM + 0338h	AWB XY Window 3 Paxel Count	AWBXY_PCNT2
CAM + 033Ch	AWB XY Window 3 R Sum	AWBXY_RSUM2
CAM + 0340h	AWB XY Window 3 G Sum	AWBXY_GSUM2
CAM + 0344h	AWB XY Window 3 B Sum	AWBXY_BSUM2
CAM + 0348h	AWB XY Window 4 Paxel Count	AWBXY_PCNT3
CAM + 034Ch	AWB XY Window 4 R Sum	AWBXY_RSUM3
CAM + 0350h	AWB XY Window 4 G Sum	AWBXY_GSUM3
CAM + 0354h	AWB XY Window 4 B Sum	AWBXY_BSUM3
CAM + 0358h	AWB XY Window 5 Paxel Count	AWBXY_PCNT4
CAM + 035Ch	AWB XY Window 5 R Sum	AWBXY_RSUM4
CAM + 0360h	AWB XY Window 5 G Sum	AWBXY_GSUM4
CAM + 0364h	AWB XY Window 5 B Sum	AWBXY_BSUM4
CAM + 0368h	AWB XY Window 6 Paxel Count	AWBXY_PCNT5
CAM + 036Ch	AWB XY Window 6 R Sum	AWBXY_RSUM5

© 2009 MediaTek Inc.

CAM + 0370h	AWB XY Window 6 G Sum	AWBXY_GSUM5
CAM + 0374h	AWB XY Window 6 B Sum	AWBXY_BSUM5
CAM + 0378h	AWB XY Window 7 Paxel Count	AWBXY_PCNT6
CAM + 037Ch	AWB XY Window 7 R Sum	AWBXY_RSUM6
CAM + 0380h	AWB XY Window 7 G Sum	AWBXY_GSUM6
CAM + 0384h	AWB XY Window 7 B Sum	AWBXY_BSUM6
CAM + 0388h	AWB XY Window 8 Paxel Count	AWBXY_PCNT7
CAM + 038Ch	AWB XY Window 8 R Sum	AWBXY_RSUM7
CAM + 0390h	AWB XY Window 8 G Sum	AWBXY_GSUM7
CAM + 0394h	AWB XY Window 8 B Sum	AWBXY_BSUM7
CAM + 0398h	AWB XY Window 9 Paxel Count	AWBXY_PCNT8
CAM + 039Ch	AWB XY Window 9 R Sum	AWBXY_RSUM8
CAM + 03A0h	AWB XY Window 9 G Sum	AWBXY_GSUM8
CAM + 03A4h	AWB XY Window 9 B Sum	AWBXY_BSUM8
CAM + 03A8h	AWB XY Window 10 Paxel Count	AWBXY_PCNT9
CAM + 03ACh	AWB XY Window 10 R Sum	AWBXY_RSUM9
CAM + 03B0h	AWB XY Window 10 G Sum	AWBXY_GSUM9
CAM + 03B4h	AWB XY Window 10 B Sum	AWBXY_BSUM9
CAM + 03B8h	AWB XY Window 11 Paxel Count	AWBXY_PCNTA
CAM + 03BCh	AWB XY Window 11 R Sum	AWBXY_RSUMA
CAM + 03C0h	AWB XY Window 11 G Sum	AWBXY_GSUMA
CAM + 03C4h	AWB XY Window 11 B Sum	AWBXY_BSUMA
CAM + 03C8h	AWB XY Window 12 Paxel Count	AWBXY_PCNTB
CAM + 03CCh	AWB XY Window 12 R Sum	AWBXY_RSUMB
CAM + 03D0h	AWB XY Window 12 G Sum	AWBXY_GSUMB
CAM + 03D4h	AWB XY Window 12 B Sum	AWBXY_BSUMB
CAM + 03D8h	Reserved	Reserved
CAM + 03DCh	Reserved	Reserved
CAM + 03E0h	Reserved	Reserved
CAM + 03E4h	Reserved	Reserved
CAM + 03E8h	Reserved	Reserved
CAM + 03ECh	Reserved	Reserved
CAM + 03F0h	Reserved	Reserved
CAM + 03F4h	Reserved	Reserved
CAM + 03F8h	Reserved	Reserved
CAM + 03FCh	Reserved	Reserved
CAM + 0400h	AF Window 1 Threshold 1 Focus Value	AF0_SUM0
CAM + 0404h	AF Window 1 Threshold 2 Focus Value	AF0_SUM1

© 2009 MediaTek Inc.

CAM + 0408h	AF Window 1 Threshold 3 Focus Value	AF0_SUM2
CAM + 040Ch	AF Window 1 Threshold 4 Focus Value	AF0_SUM3
CAM + 0410h	AF Window 1 Threshold 5 Focus Value	AF0_SUM4
CAM + 0414h	AF Window 2 Threshold 1 Focus Value	AF1_SUM0
CAM + 0418h	AF Window 2 Threshold 2 Focus Value	AF1_SUM1
CAM + 041Ch	AF Window 2 Threshold 3 Focus Value	AF1_SUM2
CAM + 0420h	AF Window 2 Threshold 4 Focus Value	AF1_SUM3
CAM + 0424h	AF Window 2 Threshold 5 Focus Value	AF1_SUM4
CAM + 0428h	AF Window 3 Threshold 1 Focus Value	AF2_SUM0
CAM + 042Ch	AF Window 3 Threshold 2 Focus Value	AF2_SUM1
CAM + 0430h	AF Window 3 Threshold 3 Focus Value	AF2_SUM2
CAM + 0434h	AF Window 3 Threshold 4 Focus Value	AF2_SUM3
CAM + 0438h	AF Window 3 Threshold 5 Focus Value	AF2_SUM4
CAM + 043Ch	AF Window 4 Threshold 1 Focus Value	AF3_SUM0
CAM + 0440h	AF Window 4 Threshold 2 Focus Value	AF3_SUM1
CAM + 0444h	AF Window 4 Threshold 3 Focus Value	AF3_SUM2
CAM + 0448h	AF Window 4 Threshold 4 Focus Value	AF3_SUM3
CAM + 044Ch	AF Window 4 Threshold 5 Focus Value	AF3_SUM4
CAM + 0450h	AF Window 5 Threshold 1 Focus Value	AF4_SUM0
CAM + 0454h	AF Window 5 Threshold 2 Focus Value	AF4_SUM1
CAM + 0458h	AF Window 5 Threshold 3 Focus Value	AF4_SUM2
CAM + 045Ch	AF Window 5 Threshold 4 Focus Value	AF4_SUM3
CAM + 0460h	AF Window 5 Threshold 5 Focus Value	AF4_SUM4
CAM + 0464h	AF Window 6 Threshold 1 Focus Value	AF5_SUM0
CAM + 0468h	AF Window 6 Threshold 2 Focus Value	AF5_SUM1
CAM + 046Ch	AF Window 6 Threshold 3 Focus Value	AF5_SUM2
CAM + 0470h	AF Window 6 Threshold 4 Focus Value	AF5_SUM3
CAM + 0474h	AF Window 6 Threshold 5 Focus Value	AF5_SUM4
CAM + 0478h	AF Window 7 Threshold 1 Focus Value	AF6_SUM0
CAM + 047Ch	AF Window 7 Threshold 2 Focus Value	AF6_SUM1
CAM + 0480h	AF Window 7 Threshold 3 Focus Value	AF6_SUM2
CAM + 0484h	AF Window 7 Threshold 4 Focus Value	AF6_SUM3
CAM + 0488h	AF Window 7 Threshold 5 Focus Value	AF6_SUM4
CAM + 048Ch	AF Window 8 Threshold 1 Focus Value	AF7_SUM0
CAM + 0490h	AF Window 8 Threshold 2 Focus Value	AF7_SUM1
CAM + 0494h	AF Window 8 Threshold 3 Focus Value	AF7_SUM2
CAM + 0498h	AF Window 8 Threshold 4 Focus Value	AF7_SUM3
CAM + 049Ch	AF Window 8 Threshold 5 Focus Value	AF7_SUM4

© 2009 MediaTek Inc.

CAM + 04A0h	AF Window 1 G Accumulation Value	AF0_GSUM
CAM + 04A4h	AF Window 2 G Accumulation Value	AF1_GSUM
CAM + 04A8h	AF Window 3 G Accumulation Value	AF2_GSUM
CAM + 04ACh	AF Window 4 G Accumulation Value	AF3_GSUM
CAM + 04B0h	AF Window 5 G Accumulation Value e	AF4_GSUM
CAM + 04B4h	AF Window 6 G Accumulation Value	AF5_GSUM
CAM + 04B8h	AF Window 7 G Accumulation Value	AF6_GSUM
CAM + 04BCh	AF Window 8 G Accumulation Value e	AF7_GSUM
CAM + 0500h	NR2 Control Register	NR2_CON
CAM + 0504h	Reserved	Reserved
CAM + 0508h	NR2 Configuration Register 2	NR2_CFG2
CAM + 050Ch	NR2 Configuration Register 3	NR2_CFG3
CAM + 0510h	NR2 Configuration Register 4	NR2_CFG4
CAM + 0514h	Reserved	Reserved
CAM + 0518h	Reserved	Reserved
CAM + 051Ch	Reserved	Reserved
CAM + 0520h	GDC Control Register	GDC_CON
CAM + 0524h	GDC Weighting Table Configuration	GDC_WTBL
CAM + 0528h	GDC Manual Curve Configuration 1	GDC_MMC1
CAM + 052Ch	GDC Manual Curve Configuration 2	GDC_MMC2
CAM + 0530h	GDC Manual Curve Configuration 3	GDC_MMC3
CAM + 0534h	GDC Manual Curve Configuration 4	GDC_MMC4
CAM + 0538h	GDC Manual Curve Configuration 5	GDC_MMC5
CAM + 053Ch	GDC Manual Curve Configuration 6	GDC_MMC6
CAM + 0540h	HST Control Register	HST_CON
CAM + 0544h	HST Configuration Register 1	HST_CFG1
CAM + 0548h	HST Configuration Register 2	HST_CFG2
CAM + 054Ch	HST Configuration Register 3	HST_CFG3
CAM + 0550h	NR1 Control Register	NR1_CON
CAM + 0554h	NR1 Defective Pixel Configuration Register 1	NR1_DP1
CAM + 0558h	NR1 Defective Pixel Configuration Register 2	NR1_DP2
CAM + 055Ch	NR1 Defective Pixel Configuration Register 3	NR1_DP3
CAM + 0560h	NR1 Defective Pixel Configuration Register 4	NR1_DP4
CAM + 0564h	NR1 Crosstalk Compensation Configuration Register	NR1_CT
CAM + 0568h	NR1 Noise Reduction Configuration Register 1	NR1_NR1
CAM + 056Ch	NR1 Noise Reduction Configuration Register 2	NR1_NR2
CAM + 0570h	NR1 Noise Reduction Configuration Register 3	NR1_NR3
CAM + 0574h	NR1 Noise Reduction Configuration Register 4	NR1_NR4

 © 2009 MediaTek Inc.

CAM + 0578h	NR1 Noise Reduction Configuration Register 5	NR1_NR5
CAM + 057Ch	NR1 Noise Reduction Configuration Register 6	NR1_NR6
CAM + 0580h	NR1 Noise Reduction Configuration Register 7	NR1_NR7
CAM + 0584h	NR1 Noise Reduction Configuration Register 8	NR1_NR8
CAM + 0588h	NR1 Noise Reduction Configuration Register 9	NR1_NR9
CAM + 058Ch	NR1 Noise Reduction Configuration Register 10	NR1_NR10
CAM + 0600h	YCCGO Control Register	YCCGO_CON
CAM + 0604h	YCCGO Configuration Register 1	YCCGO_CFG1
CAM + 0608h	YCCGO Configuration Register 2	YCCGO_CFG2
CAM + 060Ch	YCCGO Configuration Register 3	YCCGO_CFG3
CAM + 0610h	YCCGO Configuration Register 4	YCCGO_CFG4
CAM + 0614h	YCCGO Configuration Register 5	YCCGO_CFG5
CAM + 0618h	YCCGO Configuration Register 6	YCCGO_CFG6
CAM +(1000h ~ 104Ch)	AE Window Result 1~20	AEMEM(0~19)
CAM + (1050h ~ 105Ch)	AE Block Count, Bayer Size, AWB Debug 1,AWB Debug 2	
CAM + (1060h ~ 1084h)	Flare Histogram Result (1-10)	FLAREMEM(0~9)
CAM + (1088h ~ 1124h)	AF Filter (1-48)	
CAM + (1128h ~ 1144h)	AF Mean (1-8)	
CAM + (1148h ~1204h)	AWB XY Window Result(1-12) (Count, Rsum, Gsum, Bsum)	
CAM + (1208h ~ 1214h)	AWB Sum Window Result (Count, Rsum, Gsum, Bsum)	
CAM + (1218h ~ 1224h)	AWB Color Edge Window Result (Count, Rsum, Gsum, Bsum)	
CAM + (1228h ~ 1324h)	AE Histogram Result (1-64)	AEHIS(0~63)
CAM + 2000h	Reserved	Reserved
CAM + 3000h	AWB R Histogram Memory	AWBRHIS
CAM + 4000h	AWB G Histogram Memory	AWBGHIS
CAM + 5000h	AWB B Histogram Memory	AWBBHIS
CAM + 6000h	GDC Histogram Memory	GDCHIS

© 2009 MediaTek Inc.

Table 97 ***Camera Interface Register Map***

5.4.1.1 TG Register Definitions

CAM+0000h TG Phase Counter Register CAM_PHSCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCEN		CLKEN	CLKPOL	CLKCNT				CLKRS				CLKFL			
Type	R/W		R/W	R/W	R/W				R/W				R/W			
Reset	0		0	0	1				0				1			

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HVALID_EN	PXCLK_EN	PXCLK_INV	PXCLK_IN	CLKFL_POL			TGCLK_SEL	PIXCNT				DLATCH			
Type	R/W	R/W	R/W	R/W	R/W			R/W	R/W				R/W			
Reset	0	0	0	0	0			0	1				1			

PCEN TG phase counter enable control

CLKEN Enable sensor master clock (mclk) output to sensor. Note that to set sensor master clock driving, please set 0x80001500 bit[12] (add 8mA), bit[13] (add 4mA), bit[15] (slew rate fast).

CLKPOL Sensor master clock polarity control

CLKCNT Sensor master clock frequency divider control.
Sensor master clock will be 104Mhz/CLKCNT, where CLKCNT >=1.

CLKRS Sensor master clock rising edge control

CLKFL Sensor master clock falling edge control

HVALID_EN Sensor hvalid or href enable

PXCLK_EN Sensor clock input monitor.

PXCLK_INV Pixel clock inverse

PXCLK_IN Pixel clock sync enable. If sensor master based clock is Camera PLL, PXCLK_IN must be enabled. For MIPI CSI2, PXCLK_IN must be enabled, too.

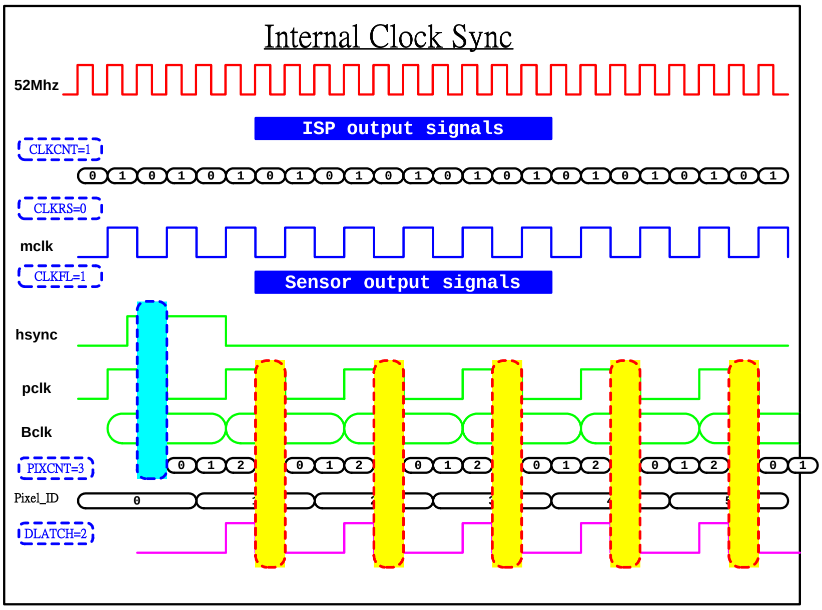
CLKFL_POL Sensor clock falling edge polarity

TGCLK_SEL Sensor master based clock selection (0: 104 Mhz, 1: Camera PLL)

PIXCNT Sensor data latch frequency control

DLATCH Sensor data latch position control
Example waveform(CLKCNT=1,CLKRS=0,CLKFL=1,PIXCNT=3,DLATCH=2**)**

 © 2009 MediaTek Inc.



CAM+0004h Sensor Size Configuration Register CAM_CAMWIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					PIXELS											
Type					R/W											
Reset					fffh											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					LINES											
Type					R/W											
Reset					fffh											

PIXEL Total input pixel number

LINE Total input line number

CAM+0008h TG Grab Range Start/End Pixel Configuration Register CAM_GRABCOL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					START											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					END											
Type					R/W											
Reset					0											

START Grab start pixel number (first pixel start from 0)

END Grab end pixel number (first pixel start from 0)

CAM+000Ch TG Grab Range Start/End Line Configuration Register CAM_GRABROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name					**START**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**END**											
Type					R/W											
Reset					0											

START Grab start line number (first line start from 0)

END Grab end line number (first line start from 0)

CAM+0010h Sensor Mode Configuration Register CAM_CSMODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											**MEMIN_DUMMYLINE[21:16]**					
Type											R/W					
Reset											48					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**CSI2_PH_SEL**	**CSI2_ECC_EN**	**DLANE4_EN**	**DLANE2_EN**	**CSI2_EN**	**VSPOL**	**HSPOL**	**PWRON**	**RST**	**AUTO**			**EN**
Type				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W
Reset				0	0	0	0	0	0	0	0	0	0			0

MEMIN_DUMMYLINE Memory input dummy line count

CSI2_PH_SEL MIPI CSI2 Packet Header selection.

0 Packet header decoded as {DI_BYTE[7:0],WC_NUMBER[7:0],WC_NUMBER[15:8]}

1 Packet header decoded as {WC_NUMBER[15:0],DI_BYTE[7:0]}

CSI2_ECC_EN MIPI CSI2 Error Correct Code enable

DLANE4_EN MIPI CSI2 4 Data Lane enable

DLANE2_EN MIPI CSI2 2 Data Lane enable

CSI2_EN MIPI CSI2 Eanble

VSPOL Sensor Vsync input polarity

HSPOL Sensor Hsync input polarity

AUTO Auto lock sensor input horizontal pixel numbers enable

EN Sensor process counter enable

CAM+0014h Component Offset Adjustment Register CAM_RGBOFF

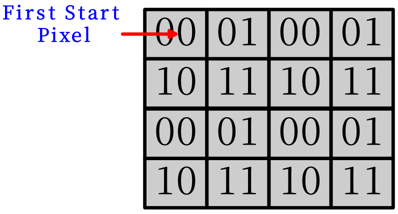
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**S00**	**OFF00**							**S01**	**OFF01**						
Type	R/W	R/W							R/W	R/W						
Reset	0	0							0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**S10**	**OFF10**							**S11**	**OFF11**						
Type	R/W	R/W							R/W	R/W						
Reset	0	0							0	0						

S00 Sign of raw data (0,0) offset adjustment control, 0 : positive 1: negative

OFF00 Raw data (0,0) offset adjustment

S01 Sign of raw data (0,1) offset adjustment control, 0 : positive 1: negative

OFF01 Raw data (0,1) offset adjustment
S10 Sign of raw data (1,0) offset adjustment control, 0 : positive 1: negative
OFF10 Raw data (1,0) offset adjustment
S11 Sign of raw data (1,1) offset adjustment control, 0 : positive 1: negative
OFF11 Raw data (1,1) offset adjustment



CAM+0018h View Finder Mode Control Register CAM_VFCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AV_SYNC_SEL	VD_INT_POL			AV_SYNC_LINENO[11:0]											
Type	R/W	R/W			R/W											
Reset	0	0			0											

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						SP_DELAY[2:0]			SP_MODE	TAKE_PIC				FR_CON		
Type						R/W			R/W	R/W				R/W		
Reset						0			0	0				0		

AV_SYNC_SEL Av_sync start point selection
- 0 Start from AV_SYNC_LINENO
- 1 Start from vsync

VD_INT_POL Vsync interrupt polarity
- 0 Vsync rising edge
- 1 Vsync Falling edge

AV_SYNC_LINENO Av_sync desired line counts

SP_DELAY[2:0] Still Picture Mode Frame Delay. When SP_DELAY is nonzero, TAKE_PIC will start to trigger after SP_DELAY Frames

SP_MODE Still Picture Mode
- **0** Preview mode, ISP will process every frame sensor send
- **1** Capture mode, ISP will only process first frame sensor send after TAKE_PIC is set

TAKE_PIC Take Picture Request

FR_CON Frame Sampling Rate Control
- **000** Every frame is sampled
- **001** One frame is sampled every 2 frames
- **010** One frame is sampled every 3 frames
- **011** One frame is sampled every 4 frames
- **100** One frame is sampled every 5 frames
- **101** One frame is sampled every 6 frames
- **110** One frame is sampled every 7 frames

© 2009 MediaTek Inc.

111 One frame is sampled every 8 frames

CAM+001Ch Camera Module Interrupt Enable Register CAM_INTEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VSYNC_INT_EN							RESZLB_EN								
Type	R/W							R/W								
Reset	0							0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								AV_SYNC_INT	FLASH_INT	ATF_INT	AEDONE	ISPDONE	IDLE	GMCOVRUN	REZOVRUN	EXPDO
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

VSYNC_INT_EN Vsync interrupt and Flash interrupt switch

0 Flash interrupt

1 Vsync interrupt

RESZLB_EN RESZ_LB (Resizer dedicate line buffer) enable. Note that CRZ and PRZ should turn off RESZ_LB when ISP use it

0 Use demo-saic line buffer as raw data dump buffer

1 Use RESZ_LB as raw data dump buffer

AV_SYNC_INT AV sync interrupt enable

FLASH_INT Flash interrupt enable, note that **VSYNC_INT_EN** switch flash and vsync interrupt

AEDONE AE done interrupt enable

ISPDONE ISP done interrupt enable

IDLE Returning idle state interrupt enable

GMCOVRUN GMC port over run interrupt enable

REZOVRUN CRZ overrun interrupt enable

EXPDO Exposure done interrupt enable

CAM+0020h Camera Module Interrupt Status Register CAM_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								AV_SYNC_INT	FLASH_INT	ATF_INT	AEDONE	ISPDONE	IDLE	GMCOVRUN	REZOVRUN	EXPDO
Type								R/W	R	R	R	R	R	R	R	R
Reset								0	0	0	0	0	0	0	0	0

AV_SYNC_INT AV sync interrupt status, occurred when desired line count equal **AV_SYNC_LINENO** in CAM_VFCON(CAM+0018h), read clear

FLASH_INT TG interrupt status, occurred when flash light pulse is done, read clear

AEDONE AE done interrupt status, occurred when 3A statistic is ok for current frame, read clear

ISPDONE ISP done interrupt status, occurred when ISP finish full frame process, read clear

IDLE Returning idle state interrupt status, occurred when ISP is in IDLE state, read clear

GMCOVRUN GMC port overrun interrupt status, occurred when ISP to memory buffer is overrun, read clear

REZOVRUN Resizer over run interrupt status, occurred when ISP to CRZ buffer is overrun, read clear

EXPDO Exposure done interrupt status, occurred when sensor send full frame, read clear

CAM+0024h Camera Module Path Config Register CAM_PATH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CNTON	CNTMODE		WRITE_LEVEL				BAYER10_OUT	REZ_DISCONN		OUTPATH_TYPE		MEMW_GBURST[2:0]			OUTPATH_EN
Type	R/W	R/W		R/W				R/W	RW		R/W		R/W			R/W
Reset	0	0		3				0	0		0		3			0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			SWAP_Y	SWAP_CBCR	INDATA_FORMAT	INTYPE_SEL			INPATH_RATE				BAYER10_IN	INPATH_LINE_DIS	INPATH_THROT_DIS	INPATH_SEL
Type		R/W	R/W	R/W	R/W	R/W			R/W				R/W	R/W	R/W	R/W
Reset		0	0	0	0	0			0				0	0	0	0

CNTON Enable Debug Mode Data Transfer Counter

CNTMODE Data Transfer Count selection

- 00 sRGB count
- 01 YCbCr count
- **Others** reserved

WRITE_LEVEL ISP output buffer level, used when ISP dump data to memory

REZ_DISCONN CRZ disconnect enable, used when ISP dump data to memory

- **0** Connect CRZ to ISP output
- **1** Disconnect CRZ to ISP output

BAYER10_OUT Bayer output format selection, note that **OUTPATH_TYPE** should be set to 00

- **0** Dump 8-bit Bayer data. For 8-bit Bayer output, memory format is **{Pixel3[7:0],Pixel2[7:0],Pixel1[7:0],Pixel0[7:0]}**
- **1** Dump 10-bit Bayer data. For 10-bit Bayer output, memory format is **{2'b0,Pixel2[9:0],Pixel1[9:0],Pixel0[9:0]}**

OUTPATH_TYPE Output path type selection

- **00** Bayer Format, note that **RESZLB_EN**(001C[24]) should be set to 1
- 01 ISP output
- 02 RGB888 Format
- 03 RGB565 Format

MEMW_GBURST Memory write burst setting. (Burst 16-4 setting is recommended)

- **00** Single access
- **03** Burst4-4 access
- **07** Burst16-4 access, only available in bayer output mode.

For Bayer 8-bit output mode, output pixel number must be multiples of 4*4, while for Bayer 10-bit output mode, output pixel number must be multiples of 3*4.

OUTPATH_EN

0 ISP dump data to memory disable

1 ISP dump data to memory enable

SWAP_Y YCbCr in Swap Y, note that **INTYPE_SEL** should be set to 001 or 101

SWAP_CBCR YCbCr in Swap Cb Cr, note that **INTYPE_SEL** should be set to 001 or 101

INDATA_FORMAT Sensor input data bit-order selection, used in ISP YUV422/YCbCr422/RGB565 format. Data input in ISP is 10-bit, datain[9:0]

0 datain[7:0] as raw data input

1 datain[9:2] as raw data input

INTYPE_SEL Input type selection

000 Bayer Format

001 YUV422 Format

101 YCbCr422 Format

010 RGB565 Format

Others Reserved

To enable YUV422/YCbCr422 input fast mode, refer to CAM + 011C bit 20

INPATH_RATE Input type rate control

BAYER10_IN Memory input path bayer data memory format

0 8-bit Bayer data, 4 bayer data packed in one word(32-bit) as (bayer3[7:0],bayer2[7:0],bayer1[7:0],bayer0[7:0])

1 10-bit Bayer data, 2 bayer data packed in one word(32-bit) as (6'b0,bayer1[9:0],6'b0,bayer0[9:0])

INPATH_LINE_DIS Input path line mode disable.

INPATH_THROT_DIS Input path throttle disable

INPATH_SEL Input path selection

0 Sensor input

1 From memory

Note that when memory input enable, Bayer output is not supported.

CAM+0028h Camera Module Input Address Register CAM_INADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CAM_INADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_INADDR[15:0]															
Type	R/W															
Reset	0															

CAM_INADDR Input memory address, used when ISP is in memory mode (**INPATH_SEL**=1)

CAM+002Ch Camera Module Output Address Register CAM_OUTADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CAM_OUTADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CAM_OUTADDR[15:0]**															
Type	R/W															
Reset	0															

CAM_OUTADDR (**OUTPATH_EN** =1) Output memory address, used when ISP dump data to memory

5.4.1.2 Color Process Register Definition

CAM+0030h Preprocessing Control Register 1 CAM_CTRL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GAIN_COMP**								**P_LIMIT**							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PGAIN_DB**		**PIXELID**				**PGAIN_INT**		**PGAIN_FRAC**							
Type	R/W		R/W				R/W		R/W							
Reset	0		0				1		0							

GAIN_COMP[7:0] Raw data compensation gain (1.7)

P_LIMIT[7:0] Edge detection input data upper bond. Edge detection applications (eg. edge ehancemnent) will not be applied on pixels those luminance above this value(0~255 range)

PGAIN_DB Pre-gain debug, set 0 for normal operation

PIXELID First Pixel type selection (Bayer data input)

- **00** **B**
- **01** **Gb**
- **02** **Gr**
- **03** **R**

PGAIN_INT[1:0] Pre-gain multiplier integer part

PGAIN_FRAC[6:0] Pre-gain multiplier fraction part

CAM+0034h AWB R,G,B Gain Control Register 1 CAM_RGBGAIN1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**B_GAIN**								
Type								R/W								
Reset								80h								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**GB_GAIN**								
Type								R/W								
Reset								80h								

B_GAIN[8:0] AWB B gain (2.7)

Gb_GAIN[8:0] AWB Gb gain (2.7)

CAM+0038h AWB R,G,B Gain Control Register 2 CAM_RGBGAIN2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**R_GAIN**								
Type								R/W								
Reset								80h								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**GR_GAIN**								
Type								R/W								
Reset								80h								

R_GAIN[8:0] AWB R gain (2.7)

GR_GAIN[8:0] AWB Gr Gain (2.7)

CAM+0044h Preprocessing Control Register 2 CAM_CTRL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**AEHIS_EN**	**AEAWG_EN**	**AECCM_EN**	**CNTEN**	**RAWACCM_MODE**	**DFDB**	**FHIS_EN**			
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Reset							0	0	0	1	0	0	0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF_EN**		**AF_SEL**					**RLEN**		**INTEN**						
Type	R/W		R/W					R/W		R/W						
Reset	1		0					0		0						

AEHIS_EN AE histogram enable

AEAWG_EN AE histogram AWB gain compensation enable
- 0 AE histogram count from shading output without AWB gain compensation
- 1 AE histogram count from shading output with AWB gain compensation

AECCM_EN AE histogram CCM enable
- 0 AE histogram count from shading output without CCM process
- 1 AE histogram count from shading output with CCM process

CNTEN AE luminance statistic window enable
- 0 AE luminance statistic window disable
- 1 AE luminance statistic window enable, only enable in bayer input mode

RAWACCM_MODE Raw data accumulation mode selection
- 0 Raw data accumulation reset when **RAWACCM_EN**(CAM+01bch) set from 0 to 1
- 1 Raw data accumulation reset every frame

DFDB Defect table debug enable, set 0 for normal operation

FHIS_EN Flare histogram enable

AF_EN Auto Focus statistic window enable, frame double buffered

AF_SEL AF Filter selection, frame double buffered
- 0 SMD filter
- 1 Tenengrad Filter

RLEN Flare histogram input selection

0 from AWB de-mosaic output only

1 from AWB de-mosaic output with window selection(CAM+0x4ch) and **AE_GAIN**(CAM+0x50h) compensation

INTEN Demo-saic internal FIFO enable, set 1 for normal operation

CAM+0048h AE Window Horizontal Width CAM_AEWINH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE_HOFFSET[6:0]							AE_WIDTH								
Type	R/W							R/W								
Reset	0							0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AE_VOFFSET[6:0]							AE_HEIGHT								
Type	R/W							R/W								
Reset	0							0								

AE_HOFFSET AE window horizontal offset pixel count, bit 7,8 refer to (CAM+0050h),frame double buffered

AE_VOFFSET AE window vertical offset line count, bit 7,8 refer to (CAM+0050h),frame double buffered

AE_WIDTH AE window block width, multiple of 2 pixels, frame double buffered

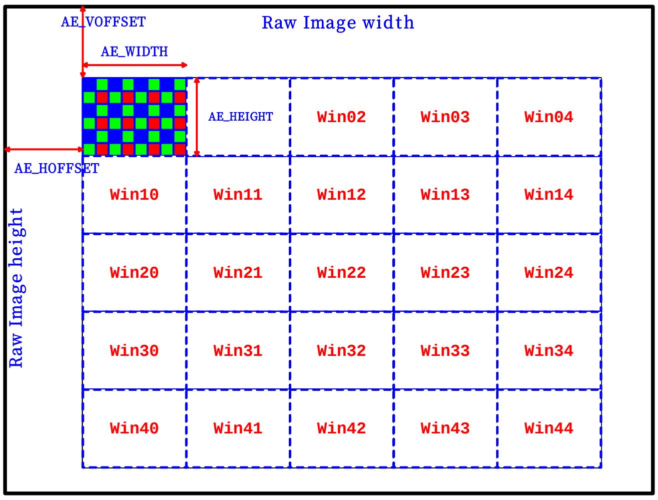
AE_HEIGHT AE window block height, multiple of 2 lines, frame double buffered

Example : if width of image is 84, then [(84/2)/5] = 8 => AE_WIDTH = 7 (count from 0)

(84/2)- 8*4 = 10 => AE_LWIDTH = 9 (count from 0)

(notes : AE Window is fixed to 5x5)

AE window partition



AE window memory allocation

CAM+0x1000	Win01[7:0],Win00[23:16]	Win00[15:0]
	Win02[15:0]	Win01[23:8]
	Win03[23:16]	Win03[7:0],Win02[23:16]
	0xae,win04[23:16]	Win04[15:0]
CAM+0x1010	Win11[7:0],Win10[23:16]	Win10[15:0]
	Win12[15:0]	Win11[23:8]
	Win13[23:16]	Win13[7:0],Win12[23:16]
	0xae,win14[23:16]	Win14[15:0]
CAM+0x1020	Win21[7:0],Win20[23:16]	Win20[15:0]
	Win22[15:0]	Win22[23:8]
	Win23[23:16]	Win23[7:0],Win22[23:16]
	0xae,win24[23:16]	Win24[15:0]
CAM+0x1030	Win31[7:0],Win30[23:16]	Win30[15:0]
	Win32[15:0]	Win32[23:8]
	Win33[23:16]	Win33[7:0],Win32[23:16]
	0xae,win34[23:16]	Win34[15:0]
CAM+0x1040	Win41[7:0],Win40[23:16]	Win40[15:0]
	Win42[15:0]	Win42[23:8]
	Win43[23:16]	Win43[7:0],Win42[23:16]
	0xae,win44[23:16]	Win44[15:0]

CAM+004Ch AE Histogram Window Vertical Height CAM_AEHISWIN

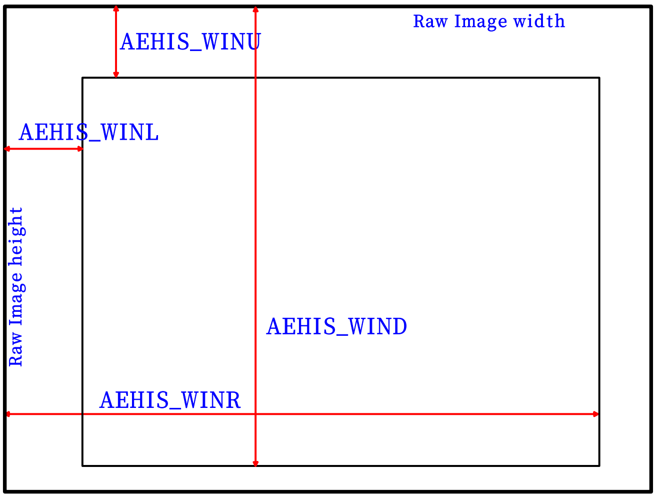
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AEHIS_WINL								AEHIS_WINR							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AEHIS_WINU								AEHIS_WIND							
Type	R/W								R/W							
Reset	0								0							

AEHIS_WINL AE Histogram window left boundary, multiples of 4 pixels

AEHIS_WINR AE Histogram window right boundary, multiples of 4 pixels

AEHIS_WINU AE Histogram window up boundary, multiples of 4 lines

AEHIS_WIND AE Histogram window down boundary, multiples of 4 lines



CAM+0050h AE Gain Register CAM_AEGAIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name						**AE_HOFFSET[8:7]**									**AE_VOFFSET[8:7]**	
Type						R/W									R/W	
Reset						0									0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**AE_GAIN**								
Type								R/W								
Reset								0								

AE_GAIN AE Histogram Gain (1.8)

CAM+0054h Global Shutter Control Register CAM_GSCTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**STROBE_POL**	**GSCTRL_EN**			**GS_EPTIME[9:0]**									
Type			R/W	R/W			R/W									
Reset			0	0			0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GS_TRTIME[7:0]**								**GS_EPTIMEU[7:0]**							
Type	R/W								R/W							
Reset	0								0							

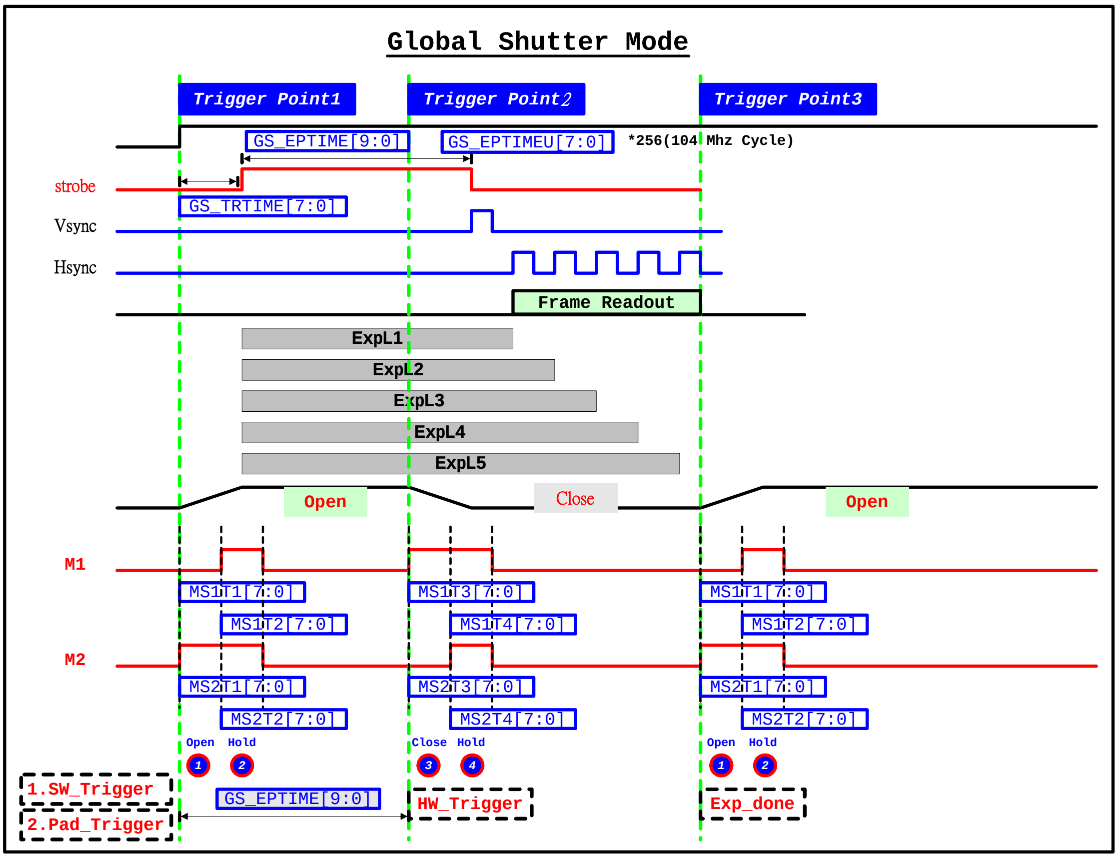
STROBE_POL Global Shutter Strobe polarity

GSCTRL_EN Global Shutter Control enable, global shutter will be triggered by **MS_SWTR** (CAM+0058h), **GSCTRL_EN** must be set first.

GS_EPTIME Global Shutter Exposure Time, time unit refer to **GS_EPTIMEU**

GS_TRTIME Global Shutter Trigger Time, Time Unit equals **MS_TIMEU**

GS_EPTIMEU Global Shutter Exposure Time = [256 * (1/104M)] * **GS_EPTIMEU**



CAM+0058h Mechanical Shutter Control Register CAM_MSCTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MECHSH0_POL	MECHSH1_POL	SINGLETR_EN	MSCTRL_EN				MS_GSMODE		MSCTRL_MODE		MS_SWTR_EN				MS_SWTR
Type	R/W	R/W	R/W	R/W				R/W		R/W		R/W				R/W
Reset	0	0	0	0				0		0		0				0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MS_TIMEU[7:0]							
Type									R/W							
Reset									0							

MECHSH0_POL Mechanical Shutter 0 Polarity

MECHSH1_POL Mechanical Shutter 1 Polarity

SINGLETR_EN Mechanical Shutter Single Trigger Enable

MSCTRL_EN Mechanical Shutter Control Enable

MS_GSMODE Mechanical Shutter Global Shutter Mode

MSCTRL_MODE Mechanical Shutter Control Mode

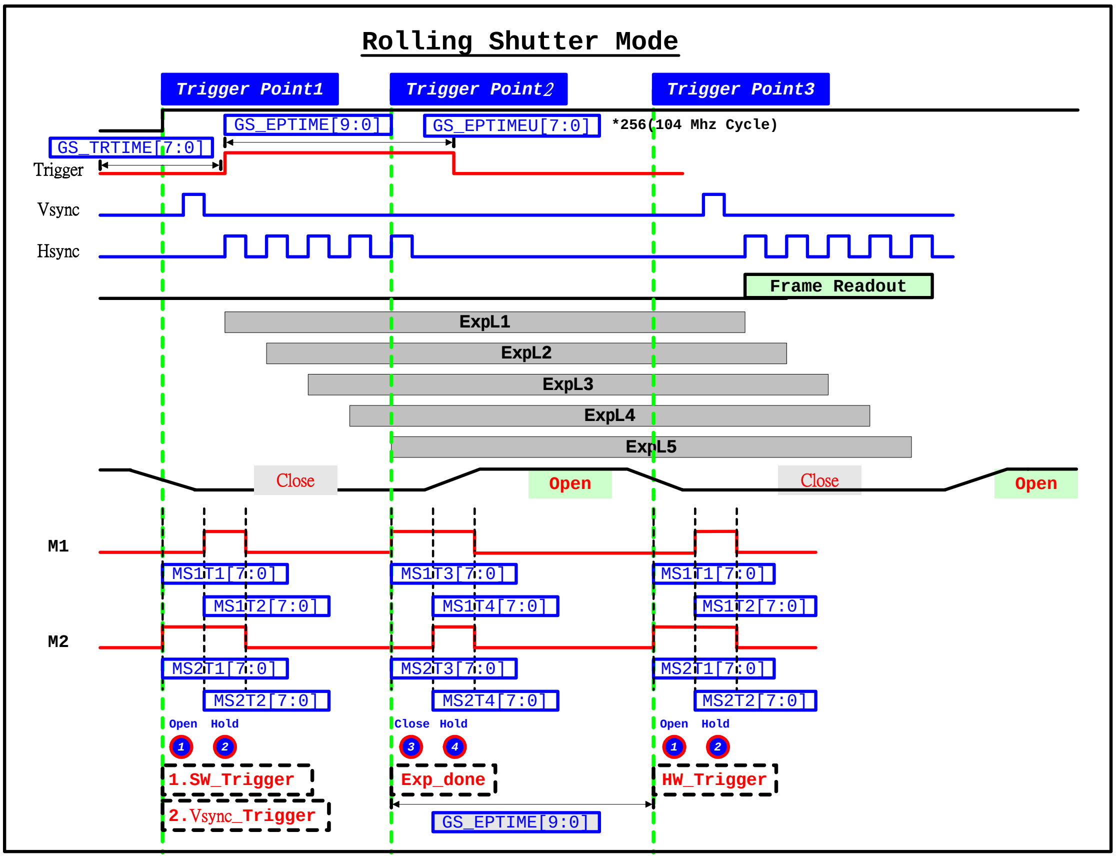
- 0 mechsh0、mechsh1 both on
- 1 mechsh0 on during phase1 and phase3, mechsh1 on during phase2
- 2 mechsh0 on during phase2,mechsh1 on during phase1 and phase3

3 simple mechanical shutter mode. Mechsh0 = 1 in trigger phase 2, 0 in phase1 and phase 3 and Mechsh1 = 0 in trigger phase 2, 1 in phase1 and phase 3

MS_SWTR_EN Mechanical Shutter Software Enable

MS_SWTR Mechanical Shutter Software Trigger

MS_TIMEU Mechanical Shutter Exposure Time Unit Setting, N. Exposure Time Unit = [4 * (1/104M)] * N



CAM+005Ch Mechanical Shutter M1 Setting Register CAM_MS1TIME

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MS1T1[7:0]								MS1T2[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MS1T3[7:0]								MS1T4[7:0]							
Type	R/W								R/W							
Reset	0								0							

MS1T1 Mechanical Shutter PAD1 Time 1

MS1T2 Mechanical Shutter PAD1 Time 2

MS1T3 Mechanical Shutter PAD1 Time 3

MS1T4 Mechanical Shutter PAD1 Time 4

CAM+0060h Mechanical Shutter M2 Setting Register CAM_MS2TIME

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MS2T1[7:0]								MS2T2[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MS2T3[7:0]								MS2T4[7:0]							
Type	R/W								R/W							
Reset	0								0							

MS2T1 Mechanical Shutter PAD2 Time 1
MS2T2 Mechanical Shutter PAD2 Time 2
MS2T3 Mechanical Shutter PAD2 Time 3
MS2T4 Mechanical Shutter PAD2 Time 4
(Note that in Phase 3, after trigger point 3, Mechanical Shutter PAD2 tied to 0 when Mechanical Shutter PAD1 waveform end)

CAM+0070h Color Processing Stage Control Register CAM_CPSCON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								DM_BPCMP_EN			HLEDGEN				VLEDGEN	DISLJ
Type								R/W			R/W				R/W	R/W
Reset								0			1				1	0

DM_BPCMP_EN Demosaic black point compensation enable
HLEDGEN Edge detection parameter, hoorizontal line edge enable
VLEDGEN Edge detection parameter, vertical line edge enable
DISLJ Demosaic parameter, line judge function in G disable

CAM+0074h Interpolation Register1 CAM_INTER1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			THRE_V						RB_SMOOTH_EN			THRE_SM				
Type			R/W						R/W			R/W				
Reset			0Ah						1			05h				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			THRE_DHV									EDGEB_RT				
Type			R/W									R/W				
Reset			19h									10h				

THRE_V Demosaic parameter
RB_SMOOTH_EN Demosaic parameter, enable RB smooth mode

THRE_SM Demosaic parameter, value increase will cause picture smoother
THRE_DHV Demosaic parameter
EDGEB_RT Edge detection parameter, edgeb threshold(2.3)

CAM+0078h Interpolation Register 2 CAM_INTER2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				LINE_RBTHD						THRE_LEDGE						
Type				R/w						R/W						
Reset				05h						14h						

LINE_RBTHD Demosaic parameter, RB line threshold
THRE_LEDGE Edge detection parameter

CAM+007Ch Edge Core Register CAM_EDGCORE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		COREH[6:0]							EMBOSS1_EN	EMBOSS2_EN	COREH2[5:0]					
Type		R/W							R/W	R/W	R/W					
Reset		08h							0	0	1Fh					
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				SPECIAL_EN	SDN_H[1:0]		SUP_H[1:0]		TOP_SLOPE	CORE_CON[6:0]						
Type				R/W	R/W		R/W		R/W	R/W						
Reset				0	2		0		0	14h						

COREH Edge detection parameter, horizontal edge core function threshold
EMBOSS1_EN Edge detection parameter, emboss effect mode 1 enable
EMBOSS2_EN Edge detection parameter, emboss effect mode 2 enable
COREH2 Edge detection parameter, horizontal edge core function parameter
SPECIAL_EN Special effect enable
SDN_H Edge detection parameter, horizontal Edge Core Function negative slope
SUP_H Edge detection parameter, horizontal Edge Core Function positive slope
TOP_SLOPE Edge detection parameter
CORE_CON Edge detection parameter

CAM+0080h Edge Gain Register 1 CAM_EDGGAIN1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SPECIGAIN		SPECIPONLY		EGAIN_H					CSI2DPHY_PWOFF		EGAIN_H2				
Type	R/W		R/W		R/W					R/W		R/W				
Reset	0		0		1					0		3				
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				EGAIN_VB					OILEN		KNEESEL		EGAINLILNE			
Type				R/W					R/W		R/W		R/W			
Reset				3					0		3		2			

SPECIGAIN[1:0] Special effect parameter, special power gain

- 00 unit gain
- 01 x2
- 10 x4
- 11 x8

SPECIPONLY Special effect parameter, p only enable

EGAIN_H[3:0] Edge detection parameter, horizontal edge gain A

CSI2DPHY_PWOFF Power down DPHY RX. Please set this value to 1 to save power when DPHY is not used.

EGAIN_H2 Edge detection parameter, horizontal edge gain B

EGAIN_VB Edge detection parameter, vertical edge gain B

OILEN Special effect parameter ,oil effect enable

KNEESEL[1:0] Edge detection parameter, edge knee threshold

EGAINLINE Edge detection parameter, edge line gain

CAM+0084h Edge Gain Register 2 CAM_EDGGAIN2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					EGAIN_VA[3:0]							EGAIN_VC[4:0]				
Type					R/W							R/W				
Reset					0							0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SPECIABS	SPECIINV		EGAIN_HC[4:0]				
Type									R/W	R/W		R/W				
Reset									0	0		Fh				

EGAIN_VA Edge detection parameter, vertical edge gain A

EGAIN_VC Edge detection parameter, vertical edge gain C

SPECIABS Special effect parameter, special absolute value enable

SPECIINV Special effect parameter, special invert enable

EGAIN_HC Edge detection parameter, horizontal edge gain C

CAM+0088h Edge Threshold Register CAM_EDGTHRE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ETH3								ETH_CON							
Type	R/W								R/W							
Reset	32h								80h							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ONLYC	THRE_EDGE_SUP								THRL_EDGE_SUP						
Type	R/W	R/W								R/W						
Reset	0	07h								07h						

ETH3 Edge detection parameter

ETH_CON Edge detection parameter

ONLYC Edge detection parameter, only c enable

THRE_EDGE_SUP Edge detection parameter
THRL_EDGE_SUP Edge detection parameter

CAM+008Ch Edge Vertical Control Register CAM_EDGVCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HPEN**	**E_TH1_V**									**HALF_V**					
Type	R/W	R/W									R/W					
Reset	0	18h									1Fh					
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**VPEN**				**SUP_V**		**SDN_V**		**E_TH3_V**							
Type	R/W				R/W		R/W		R/W							
Reset	0				0		2		32h							

HPEN Edge detection parameter, horizontal edge high pass enable
E_TH1_V Edge detection parameter
HALF_V Edge detection parameter
VPEN Edge detection parameter, vertical edge high pass enable
SUP_V Edge detection parameter, vertical edge core function positive slope
SDN_V Edge detection parameter, vertical edge core function negative slope
E_TH3_V Edge detection parameter

CAM+009Ch Color Matrix 1 Register CAM_MATRIX1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**M11**							
Type									R/W							
Reset									20h							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**M12**								**M13**							
Type	R/W								R/W							
Reset	80h								80h							

M11[7:0] Color matrix 11 value,S2.5
M12[7:0] Color matrix 12 value,S2.5
M13[7:0] Color matrix 13 value,S2.5
Process R = M11 * R + M12 * G + M13 *B

CAM+00A0h Color Matrix 2 Register CAM_MATRIX2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**M21**							
Type									R/W							
Reset									80h							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**M22**								**M23**							
Type	R/W								R/W							
Reset	20h								80h							

M21[7:0] Color matrix 21 value,S2.5
M22[7:0] Color matrix 22 value,S2.5
M23[7:0] Color matrix 23 value,S2.5
Process G = M21 * R + M22 * G + M23 *B

CAM+00A4h Color Matrix 3 Register CAM_MATRIX3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**M31**							
Type									R/W							
Reset									80h							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M32**								**M33**							
Type	R/W								R/W							
Reset	80h								20h							

M31[7:0] Color matrix 31 value,S2.5
M32[7:0] Color matrix 32 value,S2.5
M33[7:0] Color matrix 33 value,S2.5
Process B = M31 * R + M32 * G + M33 *B

CAM+00ACh Color Process Stage Control Register 2 CAM_CPSCON2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BYPGM**			**OPRGM_IVT**	**Y_EGAIN**			
Type									R/W			R/W	R/W			
Reset									1			0	2			

BYPGM Piece-wise linear gamma table bypass
- 0 piece-wise linear gamma operation enable, setting as (CAM + 01A8h ~ 01B8h)
- 1 piece-wise linear gamma operation bypass

OPDGM_IVT Piece-wise linear gamma table output inverse enable
Y_EGAIN Edge enhancement parameter, gain of edge enhancement after rgb2yuv process, available only when **YEDGE_EN** in (CAM + 0230h) set 1

CAM+00B0h Flare Gain Register CAM_FLREGAIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**FLARE_RGAIN**							
Type									R/W							
Reset									80h							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FLARE_GGAIN**								**FLARE_BGAIN**							
Type	R/W								R/W							
Reset	80h								80h							

FLARE_RGAIN Flare R gain (1.7)
FLARE_GGAIN Flare G gain (1.7)
FLARE_BGAIN Flare B gain (1.7)

CAM+00B4h Flare Offset Register CAM_FLREOFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

 © 2009 MediaTek Inc.

Name									SIGN_R		FLARE_R					
Type									R/W		R/W					
Reset									0		0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIGN_G		FLAIRE_G						SIGN_B		FLARE_B					
Type	R/W		R/W						R/W		R/W					
Reset	0		0						0		0					

SIGN_R Flare R offset sign, 0 positive, 1 negative

FLARE_R Flare R offset magnitude

SIGN_G Flare G offset sign, 0 positive, 1 negative

FLARE_G Flare G offset magnitude

SIGN_B Flare B offset sign, 0 positive, 1 negative

FLARE_B Flare B offset magnitude

CAM+00B8h Y Channel Configuration Register CAM_YCHAN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VSUP_EN		UV_LP_EN	CSUP_EDGE_GAIN				
Type									R/W		R/W	R/W				
Reset									0		0	10h				

VSUP_EN Chroma suppression enable

UV_LP_EN YUV domain uv low-pass enable

CSUP_EDGE_GAIN Chroma suppression edge gain value(1.3)

CAM+00BCh RGB2YCC Control Register RGB2YCC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YGAIN								YOFST							
Type	R/W								R/W							
Reset	FF								01							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	UGAIN								VGAIN							
Type	R/W								R/W							
Reset	90								B7							

R/G/B to Y/Cb/Cr equation is as follows

$$
\begin{aligned}
Y &= [(77R + 150G + 29B) >> 8] \times YGAIN + YOFST \\
Cb &= (B - Y) \times UGAIN + 128 \\
Cr &= (R - Y) \times VGAIN + 128
\end{aligned}
$$

CAM+00E4h MIPI CSI2 Status Register 1 CAM_CSI2STA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSI2_FRAME_NO[15:0]															

Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSI2_LINE_NO[15:0]															
Type	R															
Reset																

CSI2_FRAME_NO CSI2 Frame number. Available if sensor support additional information for frame number.

CSI2_LINE_NO CSI2 Line number. Available if sensor support additional information for line number.

CAM+00E8h MIPI CSI2 Status Register 2 CAM_CSI2STA2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSI2_WC_NUMBER[15:0]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											CSI2_DATA_TYPE[5:0]					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R					
Reset	0	0	0	0	0	0	0	0	0	0						

CSI2_WC_NUMBER CSI2 Word Count number. CSI2 packet word count.

CSI2_DATA_TYPE CSI2 Data Type. CSI2 packet data type.
Note that Raw8(**0x2A**), Raw10(**0x2B**), YUV422 8-bit(**0x1e**), RGB565 (**0x22**)

CAM+0108h Flare Histogram Result 1 CAM_HISRLT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT0[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT0[15:0]															
Type	RO															
Reset	0															

CAM_HISRLT0[21:0] Flare histogram bin 1 result

CAM+010Ch Flare Histogram Result 2 CAM_HISRLT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT1[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT1[15:0]															
Type	RO															
Reset	0															

CAM_HISRLT1[21:0] Flare histogram bin 2 result

CAM+0110h Flare Histogram Result 3 CAM_HISRLT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT2[21:16]					

Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CAM_HISRLT2[15:0]**															
Type	RO															
Reset	0															

CAM_HISRLT2[21:0] Flare histogram bin 3 result

CAM+0114h Flare Histogram Result 4 CAM_HISRLT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											**CAM_HISRLT3[21:16]**					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CAM_HISRLT3[15:0]**															
Type	RO															
Reset	0															

CAM_HISRLT3[21:0] Flare histogram bin 4 result

CAM+0118h Flare Histogram Result 5 CAM_HISRLT4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											**CAM_HISRLT4[21:16]**					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CAM_HISRLT4[15:0]**															
Type	RO															
Reset	0															

CAM_HISRLT4[21:0] Flare histogram bin 5 result

CAM+0128h Vertical Subsample Control Register CAM_VSUB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**V_SUB_EN**	**V_SUB_IN**											
Type				R/W	R/W											
Reset				0	0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**V_SUB_OUT**											
Type					R/W											
Reset					0											

V_SUB_EN Vertical sub-sample enable
V_SUB_IN Sub-sample source vertical size
V_SUB_OUT Sub-sample output vertical size

CAM+012Ch Horizontal Subsample Control Register CAM_HSUB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**H_SUB_EN**	**H_SUB_IN**											
Type				R/W	R/W											

Reset				0	0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**H_SUB_OUT**											
Type					R/W											
Reset					0											

H_SUB_EN Horizontal sub-sample enable

H_SUB_IN Sub-sample source horizontal size

H_SUB_OUT Sub-sample output horizontal size

CAM+00154h Defect Pixel Configuration Register CAM_DEFECT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**DEFECT_EN**					**DEFECT_FIFO_LEVEL**			
Type								R/W					R/W			
Reset								0					4			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

DEFECT_EN Defect table correct enable

DEFECT_FIFO_LEVEL Defect table FIFO Level, recommend value 4

CAM+0158h Defect Pixel Table Address Register CAM_DEFECT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEFFECT_ADDR[31:16]**															
Type	RW															
Reset	2000h															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DEFECT_ADDR[15:0]**															
Type	RW															
Reset	0															

DEFECT_ADDR[31:0] Defect table location address, bit0 and bit1 are fixed to 0 for word alignment

CAM+015Ch Defect Pixel Table Debug Register CAM_DEFECT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEFECT_COUNT**				**DEFFECT_YLOC[11:0]**											
Type	R				R											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**DEFECT_XLOC[11:0]**											
Type					R											
Reset																

This debug register show current status of defect table FIFO

DEFECT_COUNT Defect table counter

DEFECT_YLOC Defect table y location

DEFECT_XLOC Defect table x location

 © 2009 MediaTek Inc.

CAM+016Ch RAW Gain Control Register 1 CAM_RAWGAIN0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								RAW_RGAIN								
Type								R/W								
Reset								80h								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								RAW_GRGAIN								
Type								R/W								
Reset								80h								

RAW_RGAIN Raw R Gain (2.7), note that 0 equal unity gain

RAW_GRGAIN Raw GR Gain (2.7), note that 0 equal unity gain

CAM+0170h RAW Gain Control Register 2 CAM_RAWGAIN1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								RAW_BGAIN								
Type								R/W								
Reset								80h								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								RAW_GBGAIN								
Type								R/W								
Reset								80h								

RAW_BGAIN Raw B Gain (2.7), note that 0 equal unity gain

RAW_GBGAIN Raw GB Gain (2.7), note that 0 equal unity gain

CAM+0174h Result Window Vertical Size Register RWINV_SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				RWIN_EN	RWINV_START											
Type				R/W	R/W											
Reset				0h	0h											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					RWINV_END											
Type					R/W											
Reset					0h											

RWIN_EN Result window enable

RWINV_START Result window vertical start line

RWINV_END Result window vertical end line, end line number + 1

CAM+0178h Result Window Horizontal Size Register RWINH_SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					RWINH_START											
Type					R/W											
Reset					0h											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					RWINH_END											

Type					R/W											
Reset					0h											

RWINH_START Result window horizontal start pixel

RWINH_END Result window horizontal end pixel, end pixel number + 1

CAM+0180h Memory Input Debug Control MEMINDB_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMIN_PAUSE_EN	MEMIN_PAUSE_CONT			MEMIN_PHCOUNT											
Type	R/W	R/W			R/W											
Reset	0	0			0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MEMIN_PVCOUNT											
Type					R/W											
Reset					0											

MEMIN_PAUSE_EN Memory input pause enable

MEMIN_PAUSE_CONT Memory input pause continue, only active when pause enable is on.

MEMIN_PHCOUNT Memory input pause width count, 1 represent first pixel

MEMIN_PVCOUNT Memory input pause height count, 0 represent first line

Example : if you want to

1. stop at (0,0) : **0x80010000**
2. go to (1,0) : **0xC0020000**
3. go to (0,1) : **0x80010001 -> 0xC0010001**

CAM+0184h Memory Input Debug Register 1 MEMINDB1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					MEMIN_HCOUNT											
Type					R											
Reset					0h											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					MEMIN_VCOUNT											
Type					R											
Reset					0h											

MEMIN_HCOUNT Memory input horizontal counter

MEMIN_VCOUNT Memory input vertical counter

CAM+0188h Camera Module Debug Information Last Transfer Destination Address CAM_LASTADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LAST_ADD[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LAST_ADD[15:0]															
Type	R/W															
Reset	0															

 © 2009 MediaTek Inc.

LAST_ADD Debug Information Last Transfer Destination Address

CAM+018Ch Camera Module Frame Buffer Transfer Out Count Register CAM_XFERCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	XFER_COUNT [31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XFER_COUNT[15:0]															
Type	RO															
Reset	0															

XFER_COUNT Pixel Transfer Count per Frame

CAM+0190h Sensor Test Model Configuration Register 1 CAM_MDLCFG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VSYNC								IDLE_PIXEL_PER_LINE							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			LINECHG_EN	FULL_RANGE			ON	RST	STILL	PATTERN	PIXEL_SEL		CLK_DIV			
Type			R/W	R/W			R/W	R/W	R/W	R/W	R/W		R/W			
Reset			0	0			0	0	0	0	0		0			

VSYNC VSYNC high duration in line unit(IDLE_PIXEL_PER_LINE + PIXEL)

IDLE_PIXEL_PER_LINE HSYNC low duration in pixel unit

LINECHG_EN Pattern 0 2 lines change mode enable

FULL_RANGE Sensor Model Full Range Enable. When full range is enable, pattern data value will increase progressively every line output.

ON Enable Sensor Model.

RST Reset Sensor Model

STILL Still picture Mode

PATTERN Sensor Model Test Pattern Selection

PIXEL_SEL Sensor Model output pixel selection.

- **00** All pixels
- **01** 01 pixel
- **10** 10 pixel
- **11** 00 and 11 pixels

CLK_DIV Pixel_Clock/System_Clock Ratio

CAM +0194h Sensor Test Model Configuration Register 2 CAM_MDLCFG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						LINE										
Type						R/W										

Reset						0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**PIXEL**										
Type						R/W										
Reset						0										

LINE Sensor Model Line Number

PIXEL Sensor Model Pixel Number (HSYNC high duration in pixel unit)

CAM+01A0h CAM to CRZ interface control Register CAMCRZ_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**CAM_CRZ_INIT_EN**	**CAM_CRZ_INIT_PERIOD**							**REZ_OVRUN_FLIMIT_EN**	**REZ_OVRUN_FLIMIT_NO**			
Type	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0				0	0	0	0	0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**CAM_CRZ_FCNT_TH**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W					
Reset	0	0	0	0	0	0	0	0	0	0	0					

CAM_CRZ_ININT_EN CAM to CRZ and MDP frame initialization function enable. When this function is enable, CAM will assert frame initialization signal to MDP to indicate that frame must be initialized.

CAM_CRZ_INIT_PERIOD CAM to CRZ and MDP frame initialization period. Set CAM initialization frame signal period.

CRZ_OVRUN_FLIMIT_EN CRZ overrun frame limited enable.

CRZ_OVRUN_FLIMIT_NO CRZ overrun frame number setting. CRZ overrun interrupt will assert when **REZ_OVRUN_FLIMIT_EN** = 0 or when **REZ_OVRUN_FLIMIT_EN** = 1 and resizer overrun occurred (**REZ_OVRUN_FLIMIT_NO** + 1) frames.

CAM_CRZ_FCNT_TH CAM to CRZ FIFO count threshold .

CAM+01A8h Gamma Register 1 GMA_REG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y04**								**Y03**							
Type	R/W								R/W							
Reset	72 (0x40)								50 (0x32)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y02**								**Y01**							
Type	R/W								R/W							
Reset	32 (0x20)								20 (0x14)							

CAM+01ACh Gamma Register 2 GMA_REG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y08**								**Y07**							
Type	R/W								R/W							
Reset	120 (0x78)								110 (0x6E)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y06**								**Y05**							

Type	R/W	R/W
Reset	100 (0x64)	88 (0x58)

CAM+01B0h Gamma Register 3 GMA_REG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y12**								**Y11**							
Type	R/W								R/W							
Reset	164 (0xA4)								150 (0x96)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y10**								**Y09**							
Type	R/W								R/W							
Reset	138 (0x8A)								128 (0x80)							

CAM+01B4h Gamma Register 4 GMA_REG4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y16**								**Y15**							
Type	R/W								R/W							
Reset	206 (0xCC)								196 (0xC4)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y14**								**Y13**							
Type	R/W								R/W							
Reset	186 (0xBA)								176 (0xB0)							

CAM+01B8h Gamma Register 5 GMA_REG5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y20**								**Y19**							
Type	R/W								R/W							
Reset	248 (0xF8)								240 (0xF0)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y18**								**Y17**							
Type	R/W								R/W							
Reset	232 (0xE8)								224(0xE0)							

CAM+01BCh Raw Data Accumulation Config Register RAWACC_REG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RAWACC_EN**
Type																R/W
Reset																0

RAWACC_EN Raw Data Accumulation Enable

CAM+01C0h Raw Data Accumulation Window Register RAWWIN_REG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RAWWIN_L**								**RAWWIN_R**							
Type	R/W								R/W							
Reset	0								0							

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RAWWIN_U								RAWWIN_D							
Type	R/W								R/W							
Reset	0								0							

RAWWIN_L Raw accumulation left window
RAWWIN_R Raw accumulation right window
RAWWIN_U Raw accumulation up window
RAWWIN_D Raw accumulation down window

CAM+01C4h Raw Data Accumulation B Result RAWSUM_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RAWSUM_B[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RAWSUM_B[15:0]															
Type	R															
Reset																

RAWSUM_B Raw accumulation B channel result

CAM+01C8h Raw Data Accumulation Gb Result RAWSUM_GB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RAWSUM_GB[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RAWSUM_GB[15:0]															
Type	R															
Reset																

RAWSUM_GB Raw accumulation Gb channel result

CAM+01CCh Raw Data Accumulation Gr Result RAWSUM_GR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RAWSUM_GR[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RAWSUM_GR[15:0]															
Type	R															
Reset																

RAWSUM_GR Raw accumulation Gr channel result

CAM+01D0h Raw Data Accumulation R Result RAWSUM_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RAWSUM_R[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RAWSUM_R[15:0]
Type	R
Reset	

RAWSUM_R Raw accumulation R channel result

CAM +01D8h CAM RESET Register CAM_RESET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_CS					
Type											R					
Reset											1					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISP_FRAME_COUNT[7:0]															ISP_RESET
Type	RW															RW
Reset	0															0

CAM_CS Camera status, read only.

- **1** IDLE
- **2** VFON_IDLE
- **4** VFON_EXP
- **8** VFON_WAIT
- **16** CAPON_EXP
- **32** CAPON_DON

ISP_FRAME_COUNT ISP frame counter

ISP_RESET ISP reset. Note that reset should be assert longer than 100us to make sure GMC command done.

CAM +01DCh TG STATUS Register TG_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				SYN_VFON	LINE_COUNT[11:0]											
Type				R	R											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					PIXEL_COUNT[11:0]											
Type					R											
Reset																

SYN_VFON TG view finder status

LINE_COUNT TG line counter

PIXEL_COUNT TG pixel counter

CAM+01E0h Flash Control Register 1 FLASH_CTRL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FLASH_OUT	FLASH_CONT_FRAME[1:0]		FLASH_EN	FLASH_POL			FLASH_STARTPNT	FLASH_LNUNIT[3:0]				FLASH_LNUNIT_NO[19:16]			
Type	R	RW		RW	RW			RW	RW				RW			
Reset		0		0	0			0	0				0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	FLASH_LNUNIT_NO[15:0]
Type	RW
Reset	0

FLASH_OUT Flash out status
FLASH_CONT_FRAME Flash frame delay
FLASH_EN Flash enable
FLASH_POL Flash line polarity
FLASH_STARTPNT Flash start point
0 Start from vsync start
1 Start from expdone
FLASH_LNUNIT Flash line unit, 0~15 lines
FLASH_LNUNIT_NO Flash line unit count

CAM+01E4h Flash Control Register 2 FLASH_CTRL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FLASH_LINE[11:0]											
Type					RW											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					FLASH_PIXEL[11:0]											
Type					RW											
Reset					0											

FLASH_LINE Flash start line
FLASH_PIXEL Flash start pixel

CAM+01E8h FlashB Control Register 1 FLASHB_CTRL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FLASHB_START_FRAME[3:0]				FLASHB_LINE[11:0]											
Type	RW				RW											
Reset	0				0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		FLASHB_CONT_FRAME[2:0]			FLASHB_PIXEL[11:0]											
Type	RW				RW											
Reset	0				0											

FLASHB_START_FRAME Flash B start frame count
FLASHB_LINE Flash B start line
FLASH_CONT_FRAME Flash B continuous frame count
FLASH_PIXEL Flash B start pixel

CAM+01ECh FlashB Control Register 2 FLASHB_CTRL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FLASHB_LNUNIT[3:0]												FLASHB_LNUNIT_NO[19:16]			

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FLASH_LNUNIT_NO[15:0]															
Type	RW															
Reset	0															

FLASHB_LNUNIT Flash line unit, 0~15 lines

FLASHB_LNUNIT_NO Flash line unit count

CAM+01F0h Flare Histogram Result 6 CAM_HISRLT5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT5[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT5[15:0]															
Type	RO															
Reset	0															

CAM_HISRLT5[21:0] Flare histogram bin 6 result

CAM+01F4h Flare Histogram Result 7 CAM_HISRLT6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT6[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT6[15:0]															
Type	RO															
Reset	0															

CAM_HISRLT6[21:0] Flare histogram bin 7 result

CAM+01F8h Flare Histogram Result 8 CAM_HISRLT7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT7[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT7[15:0]															
Type	RO															
Reset	0															

CAM_HISRLT7[21:0] Flare histogram bin 8 result

CAM+01FCh Flare Histogram Result 9 CAM_HISRLT8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											CAM_HISRLT8[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAM_HISRLT8[15:0]															

 © 2009 MediaTek Inc.

Type	RO
Reset	0

CAM_HISRLT8[21:0] Flare histogram bin 9 result

CAM+0200h Flare Histogram Result 10 CAM_HISRLT9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											**CAM_HISRLT9[21:16]**					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CAM_HISRLT9[15:0]**															
Type	RO															
Reset	0															

CAM_HISRLT9[21:0] Flare histogram bin 10 result

CAM+00214h Shading Cotrol 1 Register CAM_SHADING1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**SDBLK_TRIG**	**SHADING_EN**							**SHADINGBLK_XOFFSET[5:0]**					
Type			R/W	RW							RW					
Reset			0	0							0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**SHADINGBLK_YOFFSET[5:0]**					
Type											RW					
Reset											0					

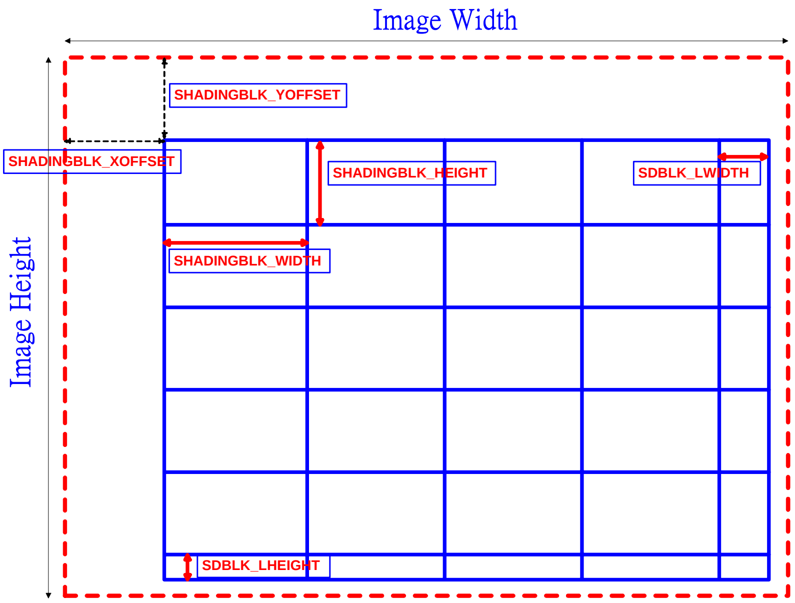
SDBLK_TRIG Shading parameter loaded trigger. When SDBLK_TRIG is set, shading block parameter will be loaded before next frame VSYNC. Note that SDBLK_TRIG and SHADING_EN must be set simultaneously, and must be close after isp_idle asserted.

SHADING_EN Shading enable

SHADINGBLK_XOFFSET Shading Block X offset

SHADINGBLK_YOFFSET Shading Block Y offset

 © 2009 MediaTek Inc.



CAM+0218h Shading Control 2 Register — CAM_SHADING 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SHADINGBLK_XNUM[3:0]				SHADINGBLK_WIDTH[11:0]											
Type	RW				RW											
Reset	6				0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SHADINGBLK_YNUM[3:0]				SHADINGBLK_HEIGHT[11:0]											
Type	RW				R/W											
Reset	4				0											

Shading configuration is based on Raw data channel (R,Gr,B,Gb). **1.**Assume Raw_Width represents Raw data channel width, and Raw_Height represents Raw data channel height, then Raw_Width = (Image_Width/2) and Raw_Height = (Image_Height/2);

2.Assume there are m block in horizontal and n block in vertical , (include CAM+0220h shading last block),

Raw_Width = m* SHADINGBLK_WIDTH + SDBLK_LWIDTH,

Raw_Height = n* SHADINGBLK_HEIGHT + SDBLK_LHEIGHT

SHADINGBLK_XNUM Shading block x number in horizontal, count from 0, eg, if there are m blocks in horizontal (include CAM+0220h shading last block) then **SHADINGBLK_XNUM = m -1.**

SHADINGBLK_YNUM Shading block y number in vertical, count from 0, eg. if there are n blocks in vertical(include CAM+0220h shading last block), then **SHADINGBLK_YNUM = n -1.**

SHADINGBLK_WIDTH Shading block width.

SHADINGBLK_HEIGHT Shading block height.

CAM+021Ch Shading Read Addr Register — SD_RADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	SHADING_RADDR[31:16]															
Type	R/W															
Reset	20h															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SHADING_RADDR[15:0]															
Type	R/W															
Reset	60h															

SHADING_RADDR[31:0] Shading coefficient read address.
Format : block0 R 12 coefficient, block0 Gr 12 coefficient, block0 Gb 12 coefficient, block0 B 12 coefficient, block1 ....

CAM+0220h Shading Last Block Config Register SD_LBLOCK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					SDBLK_LWIDTH[11:0]											
Type					R/W											
Reset					ffh											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					SDBLK_LHEIGHT[11:0]											
Type					R/W											
Reset					ffh											

SD_LWIDTH[11:0] Shading Horizontal last block width, last horizontal block can set different width.

SD_LHEIGHT[11:0] Shading Vertical last block height, last vertical block can set different height

CAM+0224h Shading Ratio Config Register SD_RATIO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			SDBLK_RATIO00								SDBLK_RATIO01					
Type			R/W								R/W					
Reset			20h								20h					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			SDBLK_RATIO10								SDBLK_RATIO11					
Type			R/W								R/W					
Reset			20h								20h					

SDBLK_RATIO00[5:0] Shading Block location 00 Ratio Gain (1.5)
SDBLK_RATIO01[5:0] Shading Block location 00 Ratio Gain (1.5)
SDBLK_RATIO10[5:0] Shading Block location 00 Ratio Gain (1.5)
SDBLK_RATIO11[5:0] Shading Block location 00 Ratio Gain (1.5)

CAM+0230h Edge Enhancement Control Register EE_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ED_GAIN_TH								CLIP_UNDER_TH							
Type	R/W								R/W							
Reset	0x20								0x64							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CLIP_OVER_TH									ED_BAND_EN		FILTER_SEL	CLIP_UNDER_EN	CLIP_OVER_EN	RGBEDGE_EN	YEDGE_EN
Type	R/W									R/W		R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

Reset	0x24									0		0	0	0	0	0

The register is for global control of edge enhancement

ED_GAIN_TH[7:0] Edge gain threshold, data range 0-255 (=32x real gain threshold)

CLIP_UNDER_TH[7:0] Undershoot clipping threshold, data range 0-127

CLIP_OVER_TH[7:0] Overshoot clipping threshold, data range 0-127

ED_BOUND_EN Edge Detection Boundary enable
- **0** Disable
- **1** Enable

FILTER_SEL "Mid-band only" or "mid-band AND high-band" in edge detection
- **0** Use mid-band detectors only
- **1** Use both mid-band AND high-band detectors

CLIP_UNDER_EN Undershoot clipping enable
- **0** Disable
- **1** Enable

CLIP_OVER_EN Overshoot clipping enable
- **0** Disable
- **1** Enable

RGBEDGE_EN EE on G pixels (after GDC)
- **0** Disable
- **1** Enable

YEDGE_EN EE on Y pixels (after RGB-to-YUV)
- 0 Disable
- 1 Enable

CAM+0234h Edge Detection LUT X Configuration ED_LUT_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**ED_LUT_X3**							
Type									R/W							
Reset									0x0A							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**ED_LUT_X2**								**ED_LUT_X1**							
Type	R/W								R/W							
Reset	0x07								0x04							

The register is for edge detection Look-Up Table (LUT) input level configuration.

ED_LUT_X1[7:0] Input level 1 for the ED LUT, data range is 0-63 (in 8-bit)

ED_LUT_X2[7:0] Input level 2 for the ED LUT, data range is 0-63 (in 8-bit)

ED_LUT_X3[7:0] Input level 3 for the ED LUT, data range is 0-63 (in 8-bit)

CAM+0238h Edge Detection LUT Y Configuration ED_LUT_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ED_LUT_Y4**								**ED_LUT_Y3**							
Type	R/W								R/W							
Reset	0x16								0x08							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**ED_LUT_Y2**								**ED_LUT_Y1**							

Type	R/W	R/W
Reset	0x04	0x02

The register is for edge detection LUT output factor configuration

ED_LUT_Y1[7:0] Output factor 1 for the ED LUT, data range is 0-63 (=32 x the real intended factor)

ED_LUT_Y2[7:0] Output factor 2 for the ED LUT, data range is 0-63 (=32 x the real intended factor)

ED_LUT_Y3[7:0] Output factor 3 for the ED LUT, data range is 0-63 (=32 x the real intended factor)

ED_LUT_Y4[7:0] Output factor 4 for the ED LUT, data range is 0-63 (=32 x the real intended factor)

CAM+024Ch AF Window 1 Register CAM_AFWIN0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 1th window left side

RIGHT[7:0] ATF 1th window right side

TOP[7:0] ATF 1th window top side

BOTTOM[7:0] ATF 1th window bottom side

CAM+0250h AF Window 2 Register CAM_AFWIN1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 2th window left side

RIGHT[7:0] ATF 2th window right side

TOP[7:0] ATF 2th window top side

BOTTOM[7:0] ATF 2th window bottom side

CAM+0254h AF Window 3 Register CAM_AFWIN2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							

Type	R/W	R/W
Reset	0	0

LEFT[7:0] ATF 3th window left side
RIGHT[7:0] ATF 3th window right side
TOP[7:0] ATF 3th window top side
BOTTOM[7:0] ATF 3th window bottom side

CAM+0258h AF Window 4 Register CAM_AFWIN3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 4th window left side
RIGHT[7:0] ATF 4th window right side
TOP[7:0] ATF 4th window top side
BOTTOM[7:0] ATF 4th window bottom side

CAM+025Ch AF Window 5 Register CAM_AFWIN4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 5th window left side
RIGHT[7:0] ATF 5th window right side
TOP[7:0] ATF 5th window top side
BOTTOM[7:0] ATF 5th window bottom side

CAM+0260h AF Threshold 1 Register CAM_AFTH0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF_TH3								AF_TH2							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF_TH1								AF_TH0							
Type	R/W								R/W							
Reset	0								0							

AF_TH0[7:0] AF Focus value threshold 1
AF_TH1[7:0] AF Focus value threshold 2
AF_TH2[7:0] AF Focus value threshold 3

AF_TH3[7:0] AF Focus value threshold 4

CAM+0264h AF Threshold 2 Register CAM_ATFTH1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AF_TH4							
Type									R/W							
Reset									0							

AF_TH4[7:0] AF Focus value threshold 5

CAM+0268h AF Window 6 Register CAM_AFWIN5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 6th window left side
RIGHT[7:0] ATF 6th window right side
TOP[7:0] ATF 6th window top side
BOTTOM[7:0] ATF 6th window bottom side

CAM+026Ch AF Window 7 Register CAM_AFWIN6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 7th window left side
RIGHT[7:0] ATF 7th window right side
TOP[7:0] ATF 7th window top side
BOTTOM[7:0] ATF 7th window bottom side

CAM+026Ch AF Window 8 Register CAM_AFWIN7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT								RIGHT							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TOP								BOTTOM							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Type	R/W								R/W							
Reset	0								0							

LEFT[7:0] ATF 8th window left side
RIGHT[7:0] ATF 8th window right side
TOP[7:0] ATF 8th window top side
BOTTOM[7:0] ATF 8th window bottom side

CAM +0274h CAM VERSION Register CAM_VERSION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YEAR[15:0]															
Type	R															
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MONTH[7:0]								DATE[7:0]							
Type	R								R							
Reset																

YEAR[15:0] Year ASCII
MONTH[7:0] Month ASCII
DATE[7:0] Date ASCII

CAM +027Ch AWB Sum Window Config Register AWBSUM_WIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBSUM_WINL								AWBSUM_WINR							
Type	R/W								R/W							
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AWBSUM_WINU								AWBSUM_WIND							
Type	R/W								R/W							
Reset																

AWBSUM_WINL[7:0] AWB sum window left corner
AWBSUM_WINR[7:0] AWB sum window right corner
AWBSUM_WINU[7:0] AWB sum window up corner
AWBSUM_WIND[7:0] AWB sum window down corner

CAM +0280h AWB Control Register AWB_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWB_EN	NEUTRAL_EN	COLOREDGE_EN	SMAREA_EN		SMAREA_NO										AWBDM_DEBUG
Type	R/W	R/W	R/W	R/W		R/W			R/W							R/W
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PAXEL_RGBH								PAXEL_YL							
Type	R/W								R/W							
Reset																

AWB_EN AWB enable
NEUTRAL_EN neural color selection enable

COLOREDGE_EN color edge selection enable
SMAREA_EN smooth area enable
SMAREA_NO[2:0] smooth area pixel count
AWBDM_DEBUG awb demosaic debug output, available when 0x80620024 bit16 set 1, and not bayer output
PAXEL_RGBH[7:0] paxel r,g,b high limit
PAXEL_YL[7:0] paxel luminance low level

CAM +0284h AWB Threshold Config AWBTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CEDGEX_TH**								**CEDGEY_TH**							
Type	R/W								R/W							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**NEUTRAL_TH**											
Type					R/W											
Reset																

CEDGEX_TH[7:0] Color Edge X threshold
CEDGEY_TH[7:0] Color Edge Y threshold
NEUTRAL_TH[11:0] Neural Flag threshold

CAM +0288h AWB Color Space H1 Config Regsiter AWBXYH1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**AWBH11**								
Type								R/W								
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**AWBH12**								
Type								R/W								
Reset																

AWBH11[8:0] AWB XY color space h11 parameter
AWBH12[8:0] AWB XY color space h12 parameter

CAM +028Ch AWB Sum Window H2 Config Register AWBXYH2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**AWBH21**								
Type								R/W								
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**AWBH22**								
Type								R/W								
Reset																

AWBH21[8:0] AWB XY color space h21 parameter
AWBH22[8:0] AWB XY color space h22 parameter

CAM +0290h AWB Color Edge Window Horizontal Config Register AWBCE_WINH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name					AWBCE_WINL											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBCE_WINR											
Type					R/W											
Reset																

AWBCE_WINL[11:0] AWB color edge window left corner
AWBCE_WINR[11:0] AWB color edge window right corner

CAM +0294h AWB Color Edge Window Vertical Config Register AWBCE_WINV

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBCE_WINU											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBCE_WIND											
Type					R/W											
Reset																

AWBCE_WINU[11:0] AWB color edge window up corner
AWBCE_WIND[11:0] AWB color edge window down corner

CAM +0298h AWB XY Window 1 Horizontal Config Register AWBXY_WINH0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINL0											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR0											
Type					R/W											
Reset																

AWBXY_WINL0[11:0] AWB color space window 1 left corner
AWBXY_WINR0[11:0] AWB color space window 1 right corner

CAM +029Ch AWB XY Window 1 Vertical Config Register AWBXY_WINV0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU0											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND0											
Type					R/W											
Reset																

AWBXY_WINU0[11:0] AWB color space window 1 up corner
AWBXY_WINL0[11:0] AWB color space window 1 down corner

CAM +02A0h AWB XY Window 2 Horizontal Config Register AWBXY_WINH1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name					AWBXY_WINL1											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR1											
Type					R/W											
Reset																

AWBXY_WINL1[11:0] AWB color space window 2 left corner
AWBXY_WINR1[11:0] AWB color space window 2 right corner

CAM +02A4h AWB XY Window 2 Vertical Config Register AWBXY_WINV1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU1											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND1											
Type					R/W											
Reset																

AWBXY_WINU1[11:0] AWB color space 2 window up corner
AWBXY_WINL1[11:0] AWB color space 2 window down corner

CAM +02A8h AWB XY Window 3 Horizontal Config Register AWBXY_WINH2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINL2											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR2											
Type					R/W											
Reset																

AWBXY_WINL2[11:0] AWB color space window 3 left corner
AWBXY_WINR2[11:0] AWB color space window 3 right corner

CAM +02ACh AWB XY Window 3 Vertical Config Register AWBXY_WINV2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU2											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND2											
Type					R/W											
Reset																

AWBXY_WINU2[11:0] AWB color space 2 window up corner
AWBXY_WINL2[11:0] AWB color space 2 window down corner

CAM +02B0h AWB XY Window 4 Horizontal Config Register AWBXY_WINH3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name					AWBXY_WINL3											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR3											
Type					R/W											
Reset																

AWBXY_WINL3[11:0] AWB color space window 4 left corner
AWBXY_WINR3[11:0] AWB color space window 4 right corner

CAM +02B4h AWB XY Window 4 Vertical Config Register AWBXY_WINV3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU3											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND3											
Type					R/W											
Reset																

AWBXY_WINU3[11:0] AWB color space 4 window up corner
AWBXY_WINL3[11:0] AWB color space 4 window down corner

CAM +02B8h AWB XY Window 5 Horizontal Config Register AWBXY_WINH4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINL4											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR4											
Type					R/W											
Reset																

AWBXY_WINL4[11:0] AWB color space window 5 left corner
AWBXY_WINR4[11:0] AWB color space window 5 right corner

CAM +02BCh AWB XY Window 5 Vertical Config Register AWBXY_WINV4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU4											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND4											
Type					R/W											
Reset																

AWBXY_WINU4[11:0] AWB color space 5 window up corner
AWBXY_WINL4[11:0] AWB color space 5 window down corner

CAM +02C0h AWB XY Window 6 Horizontal Config Register AWBXY_WINH5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name					AWBXY_WINL5											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR5											
Type					R/W											
Reset																

AWBXY_WINL5[11:0] AWB color space window 6 left corner

AWBXY_WINR5[11:0] AWB color space window 6 right corner

CAM +02C4h AWB XY Window 6 Vertical Config Register AWBXY_WINV5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU5											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND5											
Type					R/W											
Reset																

AWBXY_WINU5[11:0] AWB color space 6 window up corner

AWBXY_WINL5[11:0] AWB color space 6 window down corner

CAM +02C8h AWB XY Window 7 Horizontal Config Register AWBXY_WINH6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINL6											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR6											
Type					R/W											
Reset																

AWBXY_WINL6[11:0] AWB color space window 7 left corner

AWBXY_WINR6[11:0] AWB color space window 7 right corner

CAM +02CCh AWB XY Window 7 Vertical Config Register AWBXY_WINV6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU6											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND6											
Type					R/W											
Reset																

AWBXY_WINU6[11:0] AWB color space 7 window up corner

AWBXY_WINL6[11:0] AWB color space 7 window down corner

CAM +02D0h AWB XY Window 8 Horizontal Config Register AWBXY_WINH7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name					AWBXY_WINL7											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR7											
Type					R/W											
Reset																

AWBXY_WINL7[11:0] AWB color space window 8 left corner
AWBXY_WINR7[11:0] AWB color space window 8 right corner

CAM +02D4h AWB XY Window 8 Vertical Config Register AWBXY_WINV7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU7											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND7											
Type					R/W											
Reset																

AWBXY_WINU7[11:0] AWB color space 8 window up corner
AWBXY_WINL7[11:0] AWB color space 8 window down corner

CAM +02D8h AWB XY Window 9 Horizontal Config Register AWBXY_WINH8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINL8											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR8											
Type					R/W											
Reset																

AWBXY_WINL8[11:0] AWB color space window 9 left corner
AWBXY_WINR8[11:0] AWB color space window 9 right corner

CAM +02DCh AWB XY Window 9 Vertical Config Register AWBXY_WINV8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU8											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND8											
Type					R/W											
Reset																

AWBXY_WINU8[11:0] AWB color space 9 window up corner
AWBXY_WINL8[11:0] AWB color space 9 window down corner

CAM +02E0h AWB XY Window 10 Horizontal Config Register AWBXY_WINH9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name					AWBXY_WINL9											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINR9											
Type					R/W											
Reset																

AWBXY_WINL9[11:0] AWB color space window 10 left corner
AWBXY_WINR9[11:0] AWB color space window 10 right corner

CAM +02E4h AWB XY Window 10 Vertical Config Register AWBXY_WINV9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINU9											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WIND9											
Type					R/W											
Reset																

AWBXY_WINU9[11:0] AWB color space 10 window up corner
AWBXY_WINL9[11:0] AWB color space 10 window down corner

CAM +02E8h AWB XY Window 11 Horizontal Config Register AWBXY_WINHA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINLA											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINRA											
Type					R/W											
Reset																

AWBXY_WINLA[11:0] AWB color space window 11 left corner
AWBXY_WINRA[11:0] AWB color space window 11 right corner

CAM +02ECh AWB XY Window 11 Vertical Config Register AWBXY_WINVA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINUA											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINDA											
Type					R/W											
Reset																

AWBXY_WINUA[11:0] AWB color space 11 window up corner
AWBXY_WINLA[11:0] AWB color space 11 window down corner

CAM +02F0h AWB XY Window 12 Horizontal Config Register AWBXY_WINHB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINLB											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINRB											
Type					R/W											
Reset																

AWBXY_WINLB[11:0] AWB color space window 12 left corner
AWBXY_WINRB[11:0] AWB color space window 12 right corner

CAM +02F4h AWB XY Window 12 Vertical Config Register AWBXY_WINVB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					AWBXY_WINUB											
Type					R/W											
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					AWBXY_WINDB											
Type					R/W											
Reset																

AWBXY_WINUB[11:0] AWB color space 12 window up corner
AWBXY_WINLB[11:0] AWB color space 12 window down corner

CAM +02F8h AWB Sum Window Paxel Count AWBSUM_PCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBSUM_PCNT[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBSUM_PCNT[15:0]															
Type	R															
Reset																

AWBSUM_PCNT[21:0] AWB summation window paxel count

CAM +02FCh AWB Sum Window R Sum AWBSUM_RSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBSUM_RSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBSUM_RSUM[15:0]															
Type	R															
Reset																

AWBSUM_RSUM[28:0] AWB summation window R sum

CAM +0300h AWB Sum Window G Sum AWBSUM_GSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBSUM_GSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBSUM_GSUM[15:0]															
Type	R															
Reset																

AWBSUM_GSUM[28:0] AWB summation window G sum

CAM +0304h AWB Sum Window B Sum AWBSUM_BSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBSUM_BSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBSUM_BSUM[15:0]															
Type	R															
Reset																

AWBSUM_BSUM[28:0] AWB summation window B sum

CAM +0308h AWB Color Edge Window Paxel Count AWBCE_PCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBCE_PCNT[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBCE_PCNT[15:0]															
Type	R															
Reset																

AWBCE_PCNT[21:0] AWB color edge window paxel count

CAM +030Ch AWB Color Edge Window R Sum AWBCE_RSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBCE_RSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBCE_RSUM[15:0]															
Type	R															
Reset																

AWBCE_RSUM[28:0] AWB color edge window R sum

CAM +0310h AWB Color Edge Window G Sum AWBCE_GSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name				AWBCE_GSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBCE_GSUM[15:0]															
Type	R															
Reset																

AWBCE_GSUM[28:0] AWB color edge window G sum

CAM +0314h AWB Color Edge Window B Sum AWBCE_BSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBCE_BSUM[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBCE_BSUM[15:0]															
Type	R															
Reset																

AWBCE_BSUM[28:0] AWB color edge window B sum

CAM +0318h AWB XY Window 1 Paxel Count AWBXY_PCNT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT0[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT0[15:0]															
Type	R															
Reset																

AWBXY_PCNT0[21:0] AWB xy window paxel count

CAM +031Ch AWB XY Window 1 R Sum AWBXY_RSUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM0[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM0[15:0]															
Type	R															
Reset																

AWBXY_RSUM0[28:0] AWB xy window 1 R sum

CAM +0320h AWB XY Window 1 G Sum AWBXY_GSUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM0[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	AWBXY_GSUM0[15:0]
Type	R
Reset	

AWBXY_GSUM0[28:0] AWB xy window 1 G sum

CAM +0324h AWB XY Window 1 B Sum AWBXY_BSUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM0[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM0[15:0]															
Type	R															
Reset																

AWBXY_BSUM0[28:0] AWB xy window 1 B sum

CAM +0328h AWB XY Window 2 Paxel Count AWBXY_PCNT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT1[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT1[15:0]															
Type	R															
Reset																

AWBXY_PCNT1[21:0] AWB xy window 2 paxel count

CAM +032Ch AWB XY Window 2 R Sum AWBXY_RSUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM1[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM1[15:0]															
Type	R															
Reset																

AWBXY_RSUM1 AWB xy window 2 R sum

CAM +0330h AWB XY Window 2 G Sum AWBXY_GSUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM1[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM1[15:0]															
Type	R															

Reset	

AWBXY_GSUM1[28:0] AWB xy window 2 G sum

CAM +0334h AWB XY Window 2 B Sum AWBXY_BSUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM1[28:16]												
Type				R												
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AWBXY_BSUM1[15:0]															
Type	R															
Reset																

AWBXY_BSUM1[28:0] AWB xy window 2 B sum

CAM +0338h AWB XY Window 3 Paxel Count AWBXY_PCNT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT2[21:16]					
Type											R					
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AWBXY_PCNT2[15:0]															
Type	R															
Reset																

AWBXY_PCNT2[21:0] AWB xy window 3 paxel count

CAM +033Ch AWB XY Window 3 R Sum AWBXY_RSUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM2[28:16]												
Type				R												
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AWBXY_RSUM2[15:0]															
Type	R															
Reset																

AWBXY_RSUM2[28:0] AWB xy window 3 R sum

CAM +0340h AWB XY Window 3 G Sum AWBXY_GSUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM2[28:16]												
Type				R												
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AWBXY_GSUM2[15:0]															
Type	R															
Reset																

AWBXY_GSUM2[28:0] AWB xy window 3 G sum

CAM +0344h AWB XY Window 3 B Sum AWBXY_BSUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM2[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM2[15:0]															
Type	R															
Reset																

AWBXY_BSUM2[28:0] AWB xy window 3 B sum

CAM +0348h AWB XY Window 4 Paxel Count AWBXY_PCNT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT3[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT3[15:0]															
Type	R															
Reset																

AWBXY_PCNT3[21:0] AWB xy window 4 paxel count

CAM +034Ch AWB XY Window 4 R Sum AWBXY_RSUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM3[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM3[15:0]															
Type	R															
Reset																

AWBXY_RSUM3[28:0] AWB xy window 4 R sum

CAM +0350h AWB XY Window 4 G Sum AWBXY_GSUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM3[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM3[15:0]															
Type	R															
Reset																

AWBXY_GSUM3[28:0] AWB xy window 4 G sum

© 2009 MediaTek Inc.

CAM +0354h AWB XY Window 4 B Sum AWBXY_BSUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM3[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM3[15:0]															
Type	R															
Reset																

AWBXY_BSUM3[28:0] AWB xy window 4 B sum

CAM +0358h AWB XY Window 5 Paxel Count AWBXY_PCNT4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT4[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT4[15:0]															
Type	R															
Reset																

AWBXY_PCNT4[21:0] AWB xy window 5 paxel count

CAM +035Ch AWB XY Window 5 R Sum AWBXY_RSUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM4[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM4[15:0]															
Type	R															
Reset																

AWBXY_RSUM4[28:0] AWB xy window 5 R sum

CAM +0360h AWB XY Window 5 G Sum AWBXY_GSUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM4[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM4[15:0]															
Type	R															
Reset																

AWBXY_GSUM4[28:0] AWB xy window 5 G sum

© 2009 MediaTek Inc.

CAM +0364h AWB XY Window 5 B Sum AWBXY_BSUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM4[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM4[15:0]															
Type	R															
Reset																

AWBXY_BSUM4[28:0] AWB xy window 5 B sum

CAM +0368h AWB XY Window 6 Paxel Count AWBXY_PCNT5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT5[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT5[15:0]															
Type	R															
Reset																

AWBXY_PCNT5[21:0] AWB xy window 6 paxel count

CAM +036Ch AWB XY Window 6 R Sum AWBXY_RSUM5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM5[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM5[15:0]															
Type	R															
Reset																

AWBXY_RSUM5[28:0] AWB xy window 6 R sum

CAM +0370h AWB XY Window 6 G Sum AWBXY_GSUM5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM5[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM5[15:0]															
Type	R															
Reset																

AWBXY_GSUM5[28:0] AWB xy window 6 G sum

CAM +0374h AWB XY Window 6 B Sum AWBXY_BSUM5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM5[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM5[15:0]															
Type	R															
Reset																

AWBXY_BSUM5[28:0] AWB xy window 6 B sum

CAM +0378h AWB XY Window 7 Paxel Count AWBXY_PCNT6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT6[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT6[15:0]															
Type	R															
Reset																

AWBXY_PCNT6[21:0] AWB xy window 7 paxel count

CAM +037Ch AWB XY Window 7 R Sum AWBXY_RSUM6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM6[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM6[15:0]															
Type	R															
Reset																

AWBXY_RSUM6[28:0] AWB xy window 7 R sum

CAM +0380h AWB XY Window 7 G Sum AWBXY_GSUM6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM6[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM6[15:0]															
Type	R															
Reset																

AWBXY_GSUM6[28:0] AWB xy window 7 G sum

CAM +0384h AWB XY Window 7 B Sum AWBXY_BSUM6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM6[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM6[15:0]															
Type	R															
Reset																

AWBXY_BSUM6[28:0] AWB xy window 7 B sum

CAM +0388h AWB XY Window 8 Paxel Count AWBXY_PCNT7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT7[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT7[15:0]															
Type	R															
Reset																

AWBXY_PCNT7[21:0] AWB xy window 8 paxel count

CAM +038Ch AWB XY Window 8 R Sum AWBXY_RSUM7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM7[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM7[15:0]															
Type	R															
Reset																

AWBXY_RSUM7[28:0] AWB xy window 8 R sum

CAM +0390h AWB XY Window 8 G Sum AWBXY_GSUM7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM7[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM7[15:0]															
Type	R															
Reset																

AWBXY_GSUM7[28:0] AWB xy window 8 G sum

© 2009 MediaTek Inc.

CAM +0394h AWB XY Window 8 B Sum AWBXY_BSUM7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM7[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM7[15:0]															
Type	R															
Reset																

AWBXY_BSUM7[28:0] AWB xy window 8 B sum

CAM +0398h AWB XY Window 9 Paxel Count AWBXY_PCNT8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT8[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT8[15:0]															
Type	R															
Reset																

AWBXY_PCNT8[21:0] AWB xy window 9 paxel count

CAM +039Ch AWB XY Window 9 R Sum AWBXY_RSUM8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM8[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM8[15:0]															
Type	R															
Reset																

AWBXY_RSUM8[28:0] AWB xy window 9 R sum

CAM +03A0h AWB XY Window 9 G Sum AWBXY_GSUM8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM8[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM8[15:0]															
Type	R															
Reset																

AWBXY_GSUM8[28:0] AWB xy window 9 G sum

CAM +03A4h AWB XY Window 9 B Sum AWBXY_BSUM8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM8[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM8[15:0]															
Type	R															
Reset																

AWBXY_BSUM8[28:0] AWB xy window 9 B sum

CAM +03A8h AWB XY Window 10 Paxel Count AWBXY_PCNT9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNT9[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNT9[15:0]															
Type	R															
Reset																

AWBXY_PCNT9[21:0] AWB xy window 10 paxel count

CAM +03ACh AWB XY Window 10 R Sum AWBXY_RSUM9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUM9[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUM9[15:0]															
Type	R															
Reset																

AWBXY_RSUM9[28:0] AWB xy window 10 R sum

CAM +03B0h AWB XY Window 10 G Sum AWBXY_GSUM9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUM9[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUM9[15:0]															
Type	R															
Reset																

AWBXY_GSUM9[28:0] AWB xy window 10 G sum

CAM +03B4h AWB XY Window 10 B Sum AWBXY_BSUM9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_BSUM9[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUM9[15:0]															
Type	R															
Reset																

AWBXY_BSUM9[28:0] AWB xy window 10 B sum

CAM +03B8h AWB XY Window 11 Paxel Count AWBXY_PCNTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBXY_PCNTA[21:16]					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_PCNTA[15:0]															
Type	R															
Reset																

AWBXY_PCNTA[21:0] AWB xy window 11 paxel count

CAM +03BCh AWB XY Window 11 R Sum AWBXY_RSUMA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_RSUMA[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_RSUMA[15:0]															
Type	R															
Reset																

AWBXY_RSUMA[28:0] AWB xy window 11 R sum

CAM +03C0h AWB XY Window 11 G Sum AWBXY_GSUMA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				AWBXY_GSUMA[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_GSUMA[15:0]															
Type	R															
Reset																

AWBXY_GSUMA[28:0] AWB xy window 11 G sum

CAM +03C4h AWB XY Window 11 B Sum AWBXY_BSUMA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**AWBXY_BSUMA[28:16]**												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AWBXY_BSUMA[15:0]**															
Type	R															
Reset																

AWBXY_BSUMA[28:0] AWB xy window 11 B sum

CAM +03C8h AWB XY Window 12 Paxel Count AWBXY_PCNTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											**AWBXY_PCNTB[21:16]**					
Type											R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AWBXY_PCNTB[15:0]**															
Type	R															
Reset																

AWBXY_PCNTB[21:0] AWB xy window 12 paxel count

CAM +03CCh AWB XY Window 12 R Sum AWBXY_RSUMB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**AWBXY_RSUMB[28:16]**												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AWBXY_RSUMB[15:0]**															
Type	R															
Reset																

AWBXY_RSUMB[28:0] AWB xy window 12 R sum

CAM +03D0h AWB XY Window 12 G Sum AWBXY_GSUMB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**AWBXY_GSUMB[28:16]**												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AWBXY_GSUMB[15:0]**															
Type	R															
Reset																

AWBXY_GSUMB[28:0] AWB xy window 12 G sum

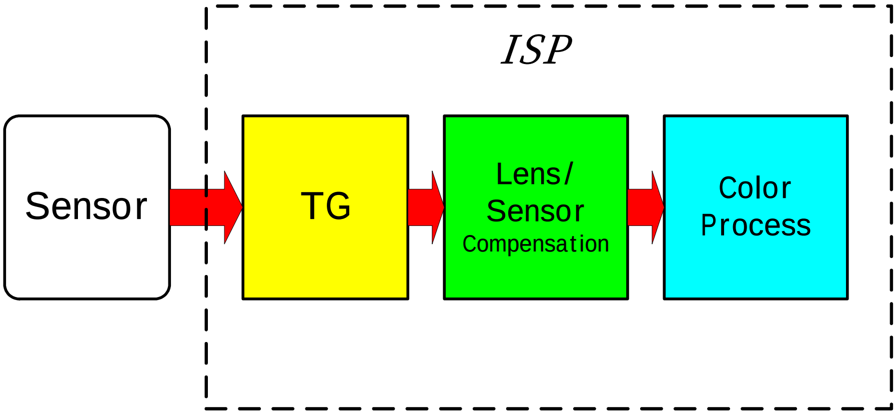
CAM +03D4h AWB XY Window 12 B Sum AWBXY_BSUMB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name				AWBXY_BSUMB[28:16]												
Type				R												
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBXY_BSUMB[15:0]															
Type	R															
Reset																

AWBXY_BSUMB[28:0] AWB xy window 12 B sum

5.5 Camera Interface



MT6516 ISP incorporates a feature rich image signal processor to connect with a variety of image sensor components. This processor consists of timing generated unit (TG) and lens/sensor compensation unit and image process unit.

Timing generated unit (TG) cooperates with master type image sensor only. That means sensor should send vertical and horizontal signals to TG. TG offers sensor required data clock and receive sensor Bayer pattern raw data by internal auto synchronization or external pixel clock synchronization. The main purpose of TG is to create data clock for master type image sensor and accept vertical/horizontal synchronization signal and sensor data, and then generate grabbed area of raw data or YUV422/RGB565 data to the lens/sensor compensation unit.

Lens/sensor compensation unit generates compensated raw data to the color process unit in Bayer raw data input mode. In YUV422/RGB565 input mode, this stage is bypassed.

Color process unit accepts Bayer pattern raw data or YUV422/RGB565 data that is generated by lens/sensor compensation unit. The output of ISP is YCbCr 888 data format which can be easily encoded by the compress engine (JPEG encoder and MPEG4 encoder). It can be the basic data domain of other data format translation such as R/G/B domain. The ISP is pipelined, and during processing stages ISP hardware can auto extract meaningful information for further AE/AF/AWB calculation. These information are temporary stored on ISP registers or memory and can be read back by MCU.

5.5.1 Register Table

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CAM + 0000h	TG Phase Counter Register	CAM_PHSCNT
CAM + 0004h	Sensor Size Configuration Register	CAM_CAMWIN

CAM + 0008h	TG Grab Range Start/End Pixel Configuration Register	CAM_GRABCOL
CAM + 000Ch	TG Grab Range Start/End Line Configuration Register	CAM_GRABROW
CAM + 0010h	Sensor Mode Configuration Register	CAM_CSMODE
CAM + 0014h	Component R, Gr, B, Gb, Offset Adjustment Register	CAM_RGBOFF
CAM + 0018h	View Finder Mode Control Register	CAM_VFCON
CAM + 001Ch	Camera Module Interrupt Enable Register	CAM_INTEN
CAM + 0020h	Camera Module Interrupt Status Register	CAM_INTSTA
CAM + 0024h	Camera Module Path Config Register	CAM_PATH
CAM + 0028h	Camera Module Input Address Register	CAM_INADDR
CAM + 002Ch	Camera Module Output Address Register	CAM_OUTADDR
CAM + 0030h	Preprocessing Control Register 1	CAM_CTRL1
CAM + 0034h	AWB R,G, B Gain Control Register 1	CAM_RGBGAIN1
CAM + 0038h	AWB R,G, B Gain Control Register 2	CAM_RGBGAIN2
CAM + 003Ch	Histogram Boundary Control Register 1	CAM_HIS0
CAM + 0040h	Histogram Boundary Control Register 2	CAM_HIS1
CAM + 0044h	Preprocessing Control Register 2	CAM_CTRL2
CAM + 0048h	AE Window 1 Register	CAM_AEWIN1
CAM + 004Ch	AE Histogram Window Register	CAM_ AEWIN2
CAM + 0050h	AE Histogram Gain Register	CAM_ AEWIN3
CAM + 0054h	Reserved	Reserved
CAM + 0058h	Reserved	Reserved
CAM + 005Ch	Reserved	Reserved
CAM + 0060h	Reserved	Reserved
CAM + 0064h	Reserved	Reserved
CAM + 0068h	Reserved	Reserved
CAM + 006Ch	AWB Window Register	CAM_AWBWIN
CAM + 0070h	Color Processing Stage Control Register	CAM_CPSCON1
CAM + 0074h	Interpolation Register 1	CAM_INTER1
CAM + 0078h	Interpolation Register 2	CAM_INTER2
CAM + 007Ch	Edge Core Register	CAM_EDGCORE
CAM + 0080h	Edge Gain Register 1	CAM_EDGGAIN1
CAM + 0084h	Edge Gain Register 2	CAM_EDGGAIN2
CAM + 0088h	Edge Threshold Register	CAM_EDGTHRE
CAM + 008Ch	Edge Vertical Control Register	CAM_EDGVCON
CAM + 0090h	Axis RGB Gain Register	CAM_AXGAIN
CAM + 0094h	AWB Configuration Register	CAM_OPDCFG
CAM + 0098h	AWB Component Parameter Register	CAM_OPDPAR
CAM + 009Ch	Color Matrix 1 Register	CAM_MATRIX1

© 2009 MediaTek Inc.

CAM + 00A0h	Color Matrix 2 Register	CAM_MATRIX2
CAM + 00A4h	Color Matrix 3 Register	CAM_MATRIX3
CAM + 00A8h	Color Matrix RGB Gain Register	CAM_MTXGAIN
CAM + 00ACh	Color Process Stage Control Register 2	CAM_CPSCON2
CAM + 00B0h	Color RGB Gain Register	CAM_CGAIN
CAM + 00B4h	Gamma RGB Flare Register	CAM_GAMFLRE
CAM + 00B8h	Y Channel Configuration Register	CAM_YCHAN
CAM + 00BCh	RGB2YCC Control Register	RGB2YCC_CON
CAM + 00C0h	Reserved	Reserved
CAM + 00C4h	Reserved	Reserved
CAM + 00C8h	Reserved	Reserved
CAM + 00CCh	Reserved	Reserved
CAM + 00D0h	Reserved	Reserved
CAM + 00D4h	AWB Y Result Register	CAM_OPDY
CAM + 00D8h	AWB MG Result Register	CAM_OPDMG
CAM + 00DCh	AWB RB Result Register	CAM_OPDRB
CAM + 00E0h	AWB Pixel Counter Register	CAM_OPDCNT
CAM + 00E4h	Reserved	Reserved
CAM + 00E8h	Reserved	Reserved
CAM + 00ECh	Reserved	Reserved
CAM + 00F0h	Reserved	Reserved
CAM + 00F4h	Reserved	Reserved
CAM + 00F8h	Reserved	Reserved
CAM + 00FCh	Reserved	Reserved
CAM + 0100h	Reserved	Reserved
CAM + 0104h	Reserved	Reserved
CAM + 0108h	Flare Histogram 1 Result	CAM_HIS_RLT0
CAM + 010Ch	Flare Histogram 2 Result	CAM_HIS_RLT1
CAM + 0110h	Flare Histogram 3 Result	CAM_HIS_RLT2
CAM + 0114h	Flare Histogram 4 Result	CAM_HIS_RLT3
CAM + 0118h	Flare Histogram 5 Result	CAM_HIS_RLT4
CAM + 011Ch	Low Pass Filter Control Register	CAM_LPFCON
CAM + 0120h	Y Low Pass Filter Control Register	CAM_YLPF
CAM + 0124h	CbCr Low Pass Filter Control Register	CAM_CLPF
CAM + 0128h	Vertical Subsample Control Register	CAM_VSUB
CAM + 012Ch	Horizontal Subsample Control Register	CAM_HSUB
CAM + 0130h	Reserved	Reserved
CAM + 0134h	Reserved	Reserved

© 2009 MediaTek Inc.

CAM + 0138h	Reserved	Reserved
CAM + 013Ch	Reserved	Reserved
CAM + 0140h	Reserved	Reserved
CAM + 0144h	Reserved	Reserved
CAM + 0148h	Reserved	Reserved
CAM + 014Ch	Reserved	Reserved
CAM + 0150h	Reserved	Reserved
CAM + 0154h	Defect Pixel Configuration Register	CAM_DEFECT0
CAM + 0158h	Defect Pixel Table Address Register	CAM_DEFECT1
CAM + 015Ch	Defect Pixel Table Debug Register	CAM_DEFECT2
CAM + 0160h	Reserved	Reserved
CAM + 0164h	Reserved	Reserved
CAM + 0168h	Reserved	Reserved
CAM + 016Ch	Raw Gain Register 1	CAM_RAWGAIN0
CAM + 0170h	Raw Gain Register 2	CAM_RAWGAIN1
CAM + 0174h	Result Window Vertical Size Register	RWINV_SEL
CAM + 0178h	Result Window Horizontal Size Register	RWINH_SEL
CAM + 017Ch	Reserved	Reserved
CAM + 0180h	Camera Interface Debug Mode Control Register	CAM_DEBUG
CAM + 0184h	Camera Module Debug Information Write Out Destination Address	CAM_DSTADDR
CAM + 0188h	Camera Module Debug Information Last Transfer Destination Address	CAM_LSTADDR
CAM + 018Ch	Camera Module Frame Buffer Transfer Out Count Register	CAM_XFERCNT
CAM + 0190h	Sensor Test Module Configuration Register 1	CAM_MDLCFG1
CAM + 0194h	Sensor Test Module Configuration Register 2	CAM_MDLCFG2
CAM + 0198h	Reserved	Reserved
CAM + 019Ch	Reserved	Reserved
CAM + 01A0h	AE Address Register	CAM_AEADDR
CAM + 01A4h	AE Window Size Register	CAM_AESIZE
CAM + 01A8h	Gamma Register 1	GMA_REG1
CAM + 01ACh	Gamma Register 2	GMA_REG2
CAM + 01B0h	Gamma Register 3	GMA_REG3
CAM + 01B4h	Gamma Register 4	GMA_REG4
CAM + 01B8h	Gamma Register 5	GMA_REG5
CAM + 01BCh	Reserved	Reserved
CAM + 01C0h	Reserved	Reserved
CAM + 01C4h	Reserved	Reserved
CAM + 01C8h	AE Area Register	CAM_AEAREA

© 2009 MediaTek Inc.

CAM + 01CCh	Reserved	Reserved
CAM + 01D0h	Reserved	Reserved
CAM + 01D4h	Flash Control Register	FLASH_CTRL
CAM + 01D8h	Cam Reset Register	CAM_RESET
CAM + 01DCh	TG Status Register	TG_STATUS
CAM + 01E0h	Reserved	Reserved
CAM + 01E4h	Reserved	Reserved
CAM + 01E8h	Reserved	Reserved
CAM + 01ECh	Flare Histogram 6 Result	CAM_HIS_RLT5
CAM + 01F0h	Flare Histogram 7 Result	CAM_HIS_RLT6
CAM + 01F4h	Flare Histogram 8 Result	CAM_HIS_RLT7
CAM + 01F8h	Flare Histogram 9 Result	CAM_HIS_RLT8
CAM + 01FCh	Flare Histogram 10 Result	CAM_HIS_RLT9
CAM + 0200h	Reserved	Reserved
CAM + 0204h	Reserved	Reserved
CAM + 0208h	Reserved	Reserved
CAM + 020Ch	Reserved	Reserved
CAM + 0210h	Reserved	Reserved
CAM + 0214h	Shading Control 1 Register	CAM_SHADING1
CAM + 0218h	Shading Control 2 Register	CAM_SHADING2
CAM + 021Ch	Shading Read Address Register	SD_RADDR
CAM + 0220h	Shading Last Block Config Register	SD_LBLOCK
CAM + 0224h	Shading Ratio Config Register	SD_RATIO
CAM + 0228h	Reserved	Reserved
CAM + 022Ch	Reserved	Reserved
CAM + 0230h	EE Control Register	EE_CTRL
CAM + 0234h	ED LUT X Configuration	ED_LUT_X
CAM + 0238h	ED LUT Y Configuration	ED_LUT_Y
CAM + 023Ch	Reserved	Reserved
CAM + 0240h	Reserved	Reserved
CAM + 0244h	Reserved	Reserved
CAM + 0248h	GMC Debug Register	CAM_GMCDEBUG
CAM + 024Ch	AF Window 1 Register	CAM_AFWIN0
CAM + 0250h	AF Window 2 Register	CAM_AFWIN1
CAM + 0254h	AF Window 3 Register	CAM_AFWIN2
CAM + 0258h	AF Window 4 Register	CAM_AFWIN3
CAM + 025Ch	AF Window 5 Register	CAM_AFWIN4
CAM + 0260h	AF Threshold 1 Register	CAM_AFTH0

 © 2009 MediaTek Inc.

CAM + 0264h	AF Threshold 2 Register	CAM_AFTH1
CAM + 0268h	Reserved	Reserved
CAM + 026Ch	Reserved	Reserved
CAM + 0270h	Reserved	Reserved
CAM + 0274h	Cam Version Register	CAM_VERSION
CAM + 027Ch	AWB Sum Window Config Register	AWBSUM_WIN
CAM + 0280h	AWB Control Register	AWB_CTRL
CAM + 0284h	AWB Threshold Config Register	AWB_TH
CAM + 0288h	AWB Color Space H1 Config Register	AWBXY_H1
CAM + 028Ch	AWB Color Space H2 Config Register	AWBXY_H2
CAM + 0290h	AWB Color Edge Window Horizontal Config Register	AWBCE_WINH
CAM + 0294h	AWB Color Edge Window Vertical Config Register	AWBCE_WINV
CAM + 0298h	AWB XY Window 1 Horizontal Config Register	AWBXY_WINH0
CAM + 029Ch	AWB XY Window 1 Vertical Config Register	AWBXY_WINV0
CAM + 02A0h	AWB XY Window 2 Horizontal Config Register	AWBXY_WINH1
CAM + 02A4h	AWB XY Window 2 Vertical Config Register	AWBXY_WINV1
CAM + 02A8h	AWB XY Window 3 Horizontal Config Register	AWBXY_WINH2
CAM + 02ACh	AWB XY Window 3 Vertical Config Register	AWBXY_WINV2
CAM + 02B0h	AWB XY Window 4 Horizontal Config Register	AWBXY_WINH3
CAM + 02B4h	AWB XY Window 4 Vertical Config Register	AWBXY_WINV3
CAM + 02B8h	AWB XY Window 5 Horizontal Config Register	AWBXY_WINH4
CAM + 02BCh	AWB XY Window 5 Vertical Config Register	AWBXY_WINV4
CAM + 02C0h	AWB XY Window 6 Horizontal Config Register	AWBXY_WINH5
CAM + 02C4h	AWB XY Window 6 Vertical Config Register	AWBXY_WINV5
CAM + 02C8h	AWB XY Window 7 Horizontal Config Register	AWBXY_WINH6
CAM + 02CCh	AWB XY Window 7 Vertical Config Register	AWBXY_WINV6
CAM + 02D0h	AWB XY Window 8 Horizontal Config Register	AWBXY_WINH7
CAM + 02D4h	AWB XY Window 8 Vertical Config Register	AWBXY_WINV7
CAM + 02D8h	AWB XY Window 9 Horizontal Config Register	AWBXY_WINH8
CAM + 02DCh	AWB XY Window 9 Vertical Config Register	AWBXY_WINV8
CAM + 02E0h	AWB XY Window 10 Horizontal Config Register	AWBXY_WINH9
CAM + 02E4h	AWB XY Window 10 Vertical Config Register	AWBXY_WINV9
CAM + 02E8h	AWB XY Window 11 Horizontal Config Register	AWBXY_WINHA
CAM + 02ECh	AWB XY Window 11 Vertical Config Register	AWBXY_WINVA
CAM + 02F0h	AWB XY Window 12 Horizontal Config Register	AWBXY_WINHB
CAM + 02F4h	AWB XY Window 12 Vertical Config Register	AWBXY_WINVB
CAM + 02F8h	AWB Sum Window Paxel Count	AWBSUM_PCNT
CAM + 02FCh	AWB Sum Window R Sum	AWBSUM_RSUM

© 2009 MediaTek Inc.

CAM + 0300h	AWB Sum Window G Sum	AWBSUM_GSUM
CAM + 0304h	AWB Sum Window B Sum	AWBSUM_BSUM
CAM + 0308h	AWB Color Edge Window Paxel Count	AWBCE_PCNT
CAM + 030Ch	AWB Color Edge Window R Sum	AWBCE_RSUM
CAM + 0310h	AWB Color Edge Window G Sum	AWBCE_GSUM
CAM + 0314h	AWB Color Edge Window B Sum	AWBCE_BSUM
CAM + 0318h	AWB XY Window 1 Paxel Count	AWBXY_PCNT0
CAM + 031Ch	AWB XY Window 1 R Sum	AWBXY_RSUM0
CAM + 0320h	AWB XY Window 1 G Sum	AWBXY_GSUM0
CAM + 0324h	AWB XY Window 1 B Sum	AWBXY_BSUM0
CAM + 0328h	AWB XY Window 2 Paxel Count	AWBXY_PCNT1
CAM + 032Ch	AWB XY Window 2 R Sum	AWBXY_RSUM1
CAM + 0330h	AWB XY Window 2 G Sum	AWBXY_GSUM1
CAM + 0334h	AWB XY Window 2 B Sum	AWBXY_BSUM1
CAM + 0338h	AWB XY Window 3 Paxel Count	AWBXY_PCNT2
CAM + 033Ch	AWB XY Window 3 R Sum	AWBXY_RSUM2
CAM + 0340h	AWB XY Window 3 G Sum	AWBXY_GSUM2
CAM + 0344h	AWB XY Window 3 B Sum	AWBXY_BSUM2
CAM + 0348h	AWB XY Window 4 Paxel Count	AWBXY_PCNT3
CAM + 034Ch	AWB XY Window 4 R Sum	AWBXY_RSUM3
CAM + 0350h	AWB XY Window 4 G Sum	AWBXY_GSUM3
CAM + 0354h	AWB XY Window 4 B Sum	AWBXY_BSUM3
CAM + 0358h	AWB XY Window 5 Paxel Count	AWBXY_PCNT4
CAM + 035Ch	AWB XY Window 5 R Sum	AWBXY_RSUM4
CAM + 0360h	AWB XY Window 5 G Sum	AWBXY_GSUM4
CAM + 0364h	AWB XY Window 5 B Sum	AWBXY_BSUM4
CAM + 0368h	AWB XY Window 6 Paxel Count	AWBXY_PCNT5
CAM + 036Ch	AWB XY Window 6 R Sum	AWBXY_RSUM5
CAM + 0370h	AWB XY Window 6 G Sum	AWBXY_GSUM5
CAM + 0374h	AWB XY Window 6 B Sum	AWBXY_BSUM5
CAM + 0378h	AWB XY Window 7 Paxel Count	AWBXY_PCNT6
CAM + 037Ch	AWB XY Window 7 R Sum	AWBXY_RSUM6
CAM + 0380h	AWB XY Window 7 G Sum	AWBXY_GSUM6
CAM + 0384h	AWB XY Window 7 B Sum	AWBXY_BSUM6
CAM + 0388h	AWB XY Window 8 Paxel Count	AWBXY_PCNT7
CAM + 038Ch	AWB XY Window 8 R Sum	AWBXY_RSUM7
CAM + 0390h	AWB XY Window 8 G Sum	AWBXY_GSUM7
CAM + 0394h	AWB XY Window 8 B Sum	AWBXY_BSUM7

© 2009 MediaTek Inc.

CAM + 0398h	AWB XY Window 9 Paxel Count	AWBXY_PCNT8
CAM + 039Ch	AWB XY Window 9 R Sum	AWBXY_RSUM8
CAM + 03A0h	AWB XY Window 9 G Sum	AWBXY_GSUM8
CAM + 03A4h	AWB XY Window 9 B Sum	AWBXY_BSUM8
CAM + 03A8h	AWB XY Window 10 Paxel Count	AWBXY_PCNT9
CAM + 03ACh	AWB XY Window 10 R Sum	AWBXY_RSUM9
CAM + 03B0h	AWB XY Window 10 G Sum	AWBXY_GSUM9
CAM + 03B4h	AWB XY Window 10 B Sum	AWBXY_BSUM9
CAM + 03B8h	AWB XY Window 11 Paxel Count	AWBXY_PCNTA
CAM + 03BCh	AWB XY Window 11 R Sum	AWBXY_RSUMA
CAM + 03C0h	AWB XY Window 11 G Sum	AWBXY_GSUMA
CAM + 03C4h	AWB XY Window 11 B Sum	AWBXY_BSUMA
CAM + 03C8h	AWB XY Window 12 Paxel Count	AWBXY_PCNTB
CAM + 03CCh	AWB XY Window 12 R Sum	AWBXY_RSUMB
CAM + 03D0h	AWB XY Window 12 G Sum	AWBXY_GSUMB
CAM + 03D4h	AWB XY Window 12 B Sum	AWBXY_BSUMB
CAM + 03D8h	Reserved	Reserved
CAM + 03DCh	Reserved	Reserved
CAM + 03E0h	Reserved	Reserved
CAM + 03E4h	Reserved	Reserved
CAM + 03E8h	Reserved	Reserved
CAM + 03ECh	Reserved	Reserved
CAM + 03F0h	Reserved	Reserved
CAM + 03F4h	Reserved	Reserved
CAM + 03F8h	Reserved	Reserved
CAM + 03FCh	Reserved	Reserved
CAM + 0400h	AF Window 1 Threshold 1 Focus Value	AF0_SUM0
CAM + 0404h	AF Window 1 Threshold 2 Focus Value	AF0_SUM1
CAM + 0408h	AF Window 1 Threshold 3 Focus Value	AF0_SUM2
CAM + 040Ch	AF Window 1 Threshold 4 Focus Value	AF0_SUM3
CAM + 0410h	AF Window 1 Threshold 5 Focus Value	AF0_SUM4
CAM + 0414h	AF Window 2 Threshold 1 Focus Value	AF1_SUM0
CAM + 0418h	AF Window 2 Threshold 2 Focus Value	AF1_SUM1
CAM + 041Ch	AF Window 2 Threshold 3 Focus Value	AF1_SUM2
CAM + 0420h	AF Window 2 Threshold 4 Focus Value	AF1_SUM3
CAM + 0424h	AF Window 2 Threshold 5 Focus Value	AF1_SUM4
CAM + 0428h	AF Window 3 Threshold 1 Focus Value	AF2_SUM0
CAM + 042Ch	AF Window 3 Threshold 2 Focus Value	AF2_SUM1

© 2009 MediaTek Inc.

CAM + 0430h	AF Window 3 Threshold 3 Focus Value	AF2_SUM2
CAM + 0434h	AF Window 3 Threshold 4 Focus Value	AF2_SUM3
CAM + 0438h	AF Window 3 Threshold 5 Focus Value	AF2_SUM4
CAM + 043Ch	AF Window 4 Threshold 1 Focus Value	AF3_SUM0
CAM + 0440h	AF Window 4 Threshold 2 Focus Value	AF3_SUM1
CAM + 0444h	AF Window 4 Threshold 3 Focus Value	AF3_SUM2
CAM + 0448h	AF Window 4 Threshold 4 Focus Value	AF3_SUM3
CAM + 044Ch	AF Window 4 Threshold 5 Focus Value	AF3_SUM4
CAM + 0450h	AF Window 5 Threshold 1 Focus Value	AF4_SUM0
CAM + 0454h	AF Window 5 Threshold 2 Focus Value	AF4_SUM1
CAM + 0458h	AF Window 5 Threshold 3 Focus Value	AF4_SUM2
CAM + 045Ch	AF Window 5 Threshold 4 Focus Value	AF4_SUM3
CAM + 0460h	AF Window 5 Threshold 5 Focus Value	AF4_SUM4
CAM + 0500h	NR2 Control Register	NR2_CON
CAM + 0504h	Reserved	Reserved
CAM + 0508h	NR2 Configuration Register 2	NR2_CFG2
CAM + 050Ch	NR2 Configuration Register 3	NR2_CFG3
CAM + 0510h	NR2 Configuration Register 4	NR2_CFG4
CAM + 0514h	Reserved	Reserved
CAM + 0518h	Reserved	Reserved
CAM + 051Ch	Reserved	Reserved
CAM + 0520h	GDC Control Register	GDC_CON
CAM + 0524h	GDC Weighting Table Configuration	GDC_WTBL
CAM + 0528h	GDC Manual Curve Configuration 1	GDC_MMC1
CAM + 052Ch	GDC Manual Curve Configuration 2	GDC_MMC2
CAM + 0530h	GDC Manual Curve Configuration 3	GDC_MMC3
CAM + 0534h	GDC Manual Curve Configuration 4	GDC_MMC4
CAM + 0538h	GDC Manual Curve Configuration 5	GDC_MMC5
CAM + 053Ch	GDC Manual Curve Configuration 6	GDC_MMC6
CAM + 0540h	HST Control Register	HST_CON
CAM + 0544h	HST Configuration Register 1	HST_CFG1
CAM + 0548h	HST Configuration Register 2	HST_CFG2
CAM + 054Ch	HST Configuration Register 3	HST_CFG3
CAM + 0550h	NR1 Control Register	NR1_CON
CAM + 0554h	NR1 Defective Pixel Configuration Register 1	NR1_DP1
CAM + 0558h	NR1 Defective Pixel Configuration Register 2	NR1_DP2
CAM + 055Ch	NR1 Defective Pixel Configuration Register 3	NR1_DP3
CAM + 0560h	NR1 Defective Pixel Configuration Register 4	NR1_DP4

CAM + 0564h	NR1 Crosstalk Compensation Configuration Register	NR1_CT
CAM + 0568h	NR1 Noise Reduction Configuration Register 1	NR1_NR1
CAM + 056Ch	NR1 Noise Reduction Configuration Register 2	NR1_NR2
CAM + 0570h	NR1 Noise Reduction Configuration Register 3	NR1_NR3
CAM + 0574h	NR1 Noise Reduction Configuration Register 4	NR1_NR4
CAM + 0578h	NR1 Noise Reduction Configuration Register 5	NR1_NR5
CAM + 057Ch	NR1 Noise Reduction Configuration Register 6	NR1_NR6
CAM + 0580h	NR1 Noise Reduction Configuration Register 7	NR1_NR7
CAM + 0584h	NR1 Noise Reduction Configuration Register 8	NR1_NR8
CAM + 0588h	NR1 Noise Reduction Configuration Register 9	NR1_NR9
CAM + 058Ch	NR1 Noise Reduction Configuration Register 10	NR1_NR10
CAM + 0600h	YCCGO Control Register	YCCGO_CON
CAM + 0604h	YCCGO Configuration Register 1	YCCGO_CFG1
CAM + 0608h	YCCGO Configuration Register 2	YCCGO_CFG2
CAM + 060Ch	YCCGO Configuration Register 3	YCCGO_CFG3
CAM + 0610h	YCCGO Configuration Register 4	YCCGO_CFG4
CAM + 0614h	YCCGO Configuration Register 5	YCCGO_CFG5
CAM + 0618h	YCCGO Configuration Register 6	YCCGO_CFG6
CAM +(1000h ~ 104Ch)	AE Window Result 1~20	AEMEM(0~19)
CAM + (1050h ~ 105Ch)	AE Block Count, Bayer Size, AWB Debug 1,AWB Debug 2	
CAM + (1060h ~ 1084h)	Flare Histogram Result (1-10)	FLAREMEM(0~9)
CAM + (1088h ~ 1124h)	AF Filter (1-48)	
CAM + (1128h ~ 1144h)	AF Mean (1-8)	
CAM + (1148h ~1204h)	AWB XY Window Result(1-12) (Count, Rsum, Gsum, Bsum)	
CAM + (1208h ~ 1214h)	AWB Sum Window Result (Count, Rsum, Gsum, Bsum)	
CAM + (1218h ~ 1224h)	AWB Color Edge Window Result (Count, Rsum, Gsum, Bsum)	
CAM + (1228h ~ 1324h)	AE Histogram Result (1-64)	AEHIS(0~63)
CAM + 2000h	Reserved	Reserved
CAM + 3000h	AWB R Histogram Memory	AWBRHIS
CAM + 4000h	AWB G Histogram Memory	AWBGHIS
CAM + 5000h	AWB B Histogram Memory	AWBBHIS
CAM + 6000h	GDC Histogram Memory	GDCHIS

© 2009 MediaTek Inc.

Table 98 ***Camera Interface Register Map***

CAM+0400h AF Window 1 Threshold 1 Focus Value AF0_SUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF0_SUM0[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF0_SUM0[15:0]**															
Type	RO															
Reset	0															

AF0_SUM0 AF Window 1 Threshold 1 Focus Value

CAM+0404h AF Window 1 Threshold 2 Focus Value AF0_SUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF0_SUM1[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF0_SUM1[15:0]**															
Type	RO															
Reset	0															

AF0_SUM1 AF Window 1 Threshold 2 Focus Value

CAM+0408h AF Window 1 Threshold 3 Focus Value AF0_SUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF0_SUM2[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF0_SUM2[15:0]**															
Type	RO															
Reset	0															

AF0_SUM2 AF Window 1 Threshold 3 Focus Value

CAM+040Ch AF Window 1 Threshold 4 Focus Value AF0_SUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF0_SUM3[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF0_SUM3[15:0]**															
Type	RO															
Reset	0															

AF0_SUM3 AF Window 1 Threshold 4 Focus Value

CAM+0410h AF Window 1 Threshold 5 Focus Value AF0_SUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	**AF0_SUM4[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF0_SUM4[15:0]**															
Type	RO															
Reset	0															

AF0_SUM4 AF Window 3 Threshold 5 Focus Value

CAM+0414h AF Window 2 Threshold 1 Focus Value AF1_SUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF1_SUM0[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF1_SUM0[15:0]**															
Type	RO															
Reset	0															

AF1_SUM0 AF Window 2 Threshold 1 Focus Value

CAM+0418h AF Window 2 Threshold 2 Focus Value AF1_SUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF1_SUM1[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF1_SUM1[15:0]**															
Type	RO															
Reset	0															

AF0_SUM1 AF Window 2 Threshold 2 Focus Value

CAM+041Ch AF Window 2 Threshold 3 Focus Value AF1_SUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF1_SUM2[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF1_SUM2[15:0]**															
Type	RO															
Reset	0															

AF1_SUM2 AF Window 2 Threshold 3 Focus Value

CAM+0420h AF Window 2 Threshold 4 Focus Value AF1_SUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF1_SUM3[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF1_SUM3[15:0]**															

Type	RO
Reset	0

AF1_SUM3 AF Window 2 Threshold 4 Focus Value

CAM+0424h AF Window 2 Threshold 5 Focus Value AF1_SUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF1_SUM4[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF1_SUM4[15:0]**															
Type	RO															
Reset	0															

AF1_SUM4 AF Window 2 Threshold 5 Focus Value

CAM+0428h AF Window 3 Threshold 1 Focus Value AF2_SUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF2_SUM0[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF2_SUM0[15:0]**															
Type	RO															
Reset	0															

AF2_SUM0 AF Window 3 Threshold 1 Focus Value

CAM+042Ch AF Window 3 Threshold 2 Focus Value AF2_SUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF2_SUM1[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF2_SUM1[15:0]**															
Type	RO															
Reset	0															

AF2_SUM1 AF Window 3 Threshold 2 Focus Value

CAM+0430h AF Window 3 Threshold 3 Focus Value AF2_SUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF2_SUM2[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF2_SUM2[15:0]**															
Type	RO															
Reset	0															

AF2_SUM2 AF Window 3 Threshold 3 Focus Value

© 2009 MediaTek Inc.

CAM+0434h AF Window 3 Threshold 4 Focus Value AF2_SUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF2_SUM3[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF2_SUM3[15:0]															
Type	RO															
Reset	0															

AF2_SUM3 AF Window 3 Threshold 4 Focus Value

CAM+0438h AF Window 3 Threshold 5 Focus Value AF2_SUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF2_SUM4[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF2_SUM4[15:0]															
Type	RO															
Reset	0															

AF2_SUM4 AF Window 3 Threshold 5 Focus Value

CAM+043Ch AF Window 4 Threshold 1 Focus Value AF3_SUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF3_SUM0[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF3_SUM0[15:0]															
Type	RO															
Reset	0															

AF3_SUM0 AF Window 4 Threshold 1 Focus Value

CAM+0440h AF Window 4 Threshold 2 Focus Value AF3_SUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF3_SUM1[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF3_SUM1[15:0]															
Type	RO															
Reset	0															

AF3_SUM1 AF Window 4 Threshold 2 Focus Value

CAM+0444h AF Window 4 Threshold 3 Focus Value AF3_SUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF3_SUM2[31:16]															
Type	RO															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF3_SUM2[15:0]**															
Type	RO															
Reset	0															

AF3_SUM2 AF Window 4 Threshold 3 Focus Value

CAM+0448h AF Window 4 Threshold 4 Focus Value AF3_SUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF3_SUM3[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF3_SUM3[15:0]**															
Type	RO															
Reset	0															

AF3_SUM3 AF Window 4 Threshold 4 Focus Value

CAM+044Ch AF Window 4 Threshold 5 Focus Value AF3_SUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF3_SUM4[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF3_SUM4[15:0]**															
Type	RO															
Reset	0															

AF3_SUM4 AF Window 4 Threshold 5 Focus Value

CAM+0450h AF Window 5 Threshold 1 Focus Value AF4_SUM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF4_SUM0[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF4_SUM0[15:0]**															
Type	RO															
Reset	0															

AF4_SUM0 AF Window 5 Threshold 1 Focus Value

CAM+0454h AF Window 5 Threshold 2 Focus Value AF4_SUM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AF4_SUM1[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AF4_SUM1[15:0]**															
Type	RO															
Reset	0															

© 2009 MediaTek Inc.

AF4_SUM1 AF Window 5 Threshold 2 Focus Value

CAM+0458h AF Window 5 Threshold 3 Focus Value AF4_SUM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF4_SUM2[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF4_SUM2[15:0]															
Type	RO															
Reset	0															

AF4_SUM2 AF Window 5 Threshold 3 Focus Value

CAM+045Ch AF Window 5 Threshold 4 Focus Value AF4_SUM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF4_SUM3[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF4_SUM3[15:0]															
Type	RO															
Reset	0															

AF4_SUM3 AF Window 5 Threshold 4 Focus Value

CAM+0460h AF Window 5 Threshold 5 Focus Value AF4_SUM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AF4_SUM4[31:16]															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AF4_SUM4[15:0]															
Type	RO															
Reset	0															

AF4_SUM4 AF Window 5 Threshold 5 Focus Value

CAM+0500h NR2 Control Register NR2_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															ENC	ENY
Type															R/W	R/W
Reset															0	0

ENC Enable bit of chroma (U/V) channel noise reduction

- **0** Disable
- **1** Enable

ENY Enable bit of luma (Y) channel noise reduction

- **0** Disable
- **1** Enable

CAM+0508h NR2 Configuration Register 2 NR2_CFG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									UV_SMPL		S2			S3		
Type									R/W		R/W			R/W		
Reset									2		2			4		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SY1				SC1				GNY				GNC			
Type	R/W				R/W				R/W				R/W			
Reset	2				2				4				4			

UV_SMPL Mode selection of U/V down-sample

- **00** Right
- **01** Left
- **10** Average

S2 Weighting factor S2. (S2 is valid from 0 to 4. Normalized gain = S2/4)

S3 Weighting factor S3. (S3 is valid from 0 to 4. Normalized gain = S3/4)

SY1 Weighting factor SY1. (SY1 is valid from 0 to 8. The normalized gain = SY1/4.)

SC1 Weighting factor SC1. (SC1 is valid from 0 to 8. The normalized gain = SC1/4.)

GNY $Gain_Y$ of luma channel. (GNY is valid from 0 to 8. Normalized gain = GNY/8.)

GNC $Gain_c$ of chroma channel. (GNC is valid from 0 to 8. Normalized gain = GNC/8.)

CAM+050Ch NR2 Configuration Register 3 NR2_CFG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PTY1								PTY2							
Type	R/w								R/w							
Reset	2								4							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PTY3								PTY4							
Type	R/w								R/w							
Reset	6								8							

PTY1 Y1 point. (0-255)

PTY2 Y2 point. (0-255)

PTY3 Y3 point. (0-255)

PTY4 Y4 point. (0-255)

Note: The four values are valid from 0 to 255, and PTY1<=PTY2<=PTY3<=PTY4

CAM+0510h NR2 Configuration Register 4 NR2_CFG4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PTC1								PTC2							
Type	R/w								R/w							
Reset	4								6							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PTC3								PTC4							
Type	R/w								R/w							

Reset	8	10

PTC1 C1 point.
PTC2 C2 point.
PTC3 C3 point.
PTC4 C4 point.
Note: The four values are valid from 0 to 255, and PTC1<=PTC2<=PTC3<=PTC4

CAM+0520h GDC Control Register GDC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PNUM			MODE	LPFEN	EN
Type											R/W			R/W	R/W	R/W
Reset											0			1	1	0

PNUM Selection of the pixel number of image histogram. PNUM should be consistent with HST setting. The register should be configured correctly when auto mode is selected

MODE Mode of GDC operation
- **0** Auto
- **1** Manual

LPFEN Enable bit of 1-D LPF function
- **0** Disable
- **1** Enable

EN Enable bit of GDC function
- **0** Disable
- **1** Enable

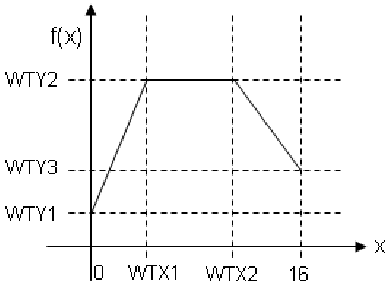
CAM+0524h GDC Weighting Table Configuration GDC_WTBL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			WTY1								WTY2					
Type			R/W								R/W					
Reset			2								12					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			WTY3						WTX1				WTX2			
Type			R/W						R/W				R/W			
Reset			4						4				13			

The register is for the configuration of the weighting table f(x).

WTY1, WTY2, WTY3 Data is valid from 0 to 63. The normalized gain is WTY/64.

WTX1, WTX2 Data is valid from 0 to 15. The scaled value is WTX * 256 in 12-bit data domain.



 © 2009 MediaTek Inc.

CAM+0528h GDC Manual Curve Configuration 1 GDC_MMC1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X01**								**X02**							
Type	R/W								R/W							
Reset	4								8							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X03**								**X04**							
Type	R/W								R/W							
Reset	16								24							

CAM+052Ch GDC Manual Curve Configuration 2 GDC_MMC2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X05**								**X06**							
Type	R/W								R/W							
Reset	32								48							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X07**								**X08**							
Type	R/W								R/W							
Reset	64								96							

CAM+0530h GDC Manual Curve Configuration 3 GDC_MMC3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X09**								**X10**							
Type	R/W								R/W							
Reset	128								160							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X11**								**X12**							
Type	R/W								R/W							
Reset	192								224							

CAM+0534h GDC Manual Curve Configuration 4 GDC_MMC4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y01**								**Y02**							
Type	R/W								R/W							
Reset	4								8							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y03**								**Y04**							
Type	R/W								R/W							
Reset	16								24							

CAM+0538h GDC Manual Curve Configuration 5 GDC_MMC5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y05**								**Y06**							
Type	R/W								R/W							
Reset	32								48							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y07**								**Y08**							
Type	R/W								R/W							
Reset	64								96							

The register is used to set the mapping curve while operating in Manual mode.

CAM+053Ch GDC Manual Curve Configuration 6 GDC_MMC6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y09**								**Y10**							
Type	R/W								R/W							
Reset	128								160							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y11**								**Y12**							
Type	R/W								R/W							
Reset	192								224							

The register is used to set the mapping curve while operating in Manual mode. The following figure shows an example of the mapping curve.

CAM+0540h HST Control Register HST_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**AUTO**	**MODE**			**EN**
Type												R/W	R/W			R/W
Reset												0	0			0

AUTO Hardware auto sub-sample for histogram.

- 0 Disable. The sample point is calculated according to HST_CFG1, HST_CFG2 and HST_CFG3.
- 1 Auto resize, HST_CFG3 is meaningless. Hardware will auto sub-sample points for histogram to fit GDC_CON.PNUM. Note that the window of histogram defined in HST_CFG1 and HST_CFG2 should be larger than GDC_CON.PNUM.

MODE Select the type of image source

- **0** Calculate the Y histogram (Y=R/4+G/2+B/4)
- **1** Calculate the histogram of R only
- **2** Calculate the histogram of G only
- **3** Calculate the histogram of B only

EN Enable bit of HST function.

- **0** Disable
- **1** Enable

CAM+0544h HST Configuration Register 1 HST_CFG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**STA_H**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**END_H**											
Type					R/W											
Reset					639											

STA_H The position of the starting pixel of each row
END_H The position of the ending pixel of each row

CAM+0548h HST Configuration Register 2 HST_CFG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					STA_V											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					END_V											
Type					R/W											
Reset					479											

The register is for global configuration of HST.

STA_V The position of the starting pixel of each column
END_V The position of the ending pixel of each column

CAM+054Ch HST Configuration Register 3 HST_CFG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			SKP_H								SKP_V					
Type			R/W								R/W					
Reset			3								3					

The register is for global configuration of HST.

SKP_H Down-sample ratio of each row (e.g. if the down-sample ratio = 1/n, SKP_H should be set to n-1)
SKP_V Down-sample ratio of each column

CAM+0550h NR1 Control Register NR1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BWD										BHT					
Type	R/W										R/W					
Reset	0										0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BHT								MDCT		BINOP	BINMD	ENBIN	ENNR	ENCT	ENDP
Type	R/W								R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0								3		0	0	0	0	0	0

BWD[9:0] The register is used to set the image width of NR1 output in 3x3 binning mode
BHT [9:0] The register is used to set the image height of NR1 output in 3x3 binning mode
MDCT The register bit is used to configure the mode of crosstalk compensation
- **0** Use noise reduction table to determine on or off
- **3** Use threshold to determine on or off

BINOP The register bit is used to select the method of pixel binning
- **0** Accumulation

1 Average

BINMD The register bit is used to select the size of pixel binning

0 2x2 pixel binning

1 3x3 pixel binning

ENBIN Enable bit of pixel binning

0 Disable

1 Enable **(Note: when the bit is '1', functions of noise reduction, crosstalk and auto defect pixel detection are turned off automatically.)**

ENNR Enable bit of noise reduction

0 Disable

1 Enable **(Note: when the bit is '1', the output resolution will decrease by 2 horizontally and vertically.)**

ENCT Enable bit of crosstalk compensation

0 Disable

1 Enable

ENDP Enable bit of auto defective pixel detection & correction

0 Disable

1 Enable

Note: The input width and height is not the same with output width and height if NR or BINNING is enabled. And the input size has certain kind of restriction. Please refer to the following table.

	Input (width and height)	(
BINNING 3x3	6n	2
BINNING 2x2	4n+2	2
NR	n+2	n

Ps. N is an integer

CAM+0554h NR1 Defective Pixel Configuration Register 1 NR1_DP1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DP_THRD2						DP_THRD0									
Type	R/W						R/W									
Reset	5						40									
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DP_THRD3						DP_THRD1									
Type	R/W						R/W									
Reset	5						80									

The register is for global configuration of defective pixel detection.

DP_THRD2 'threshold2'. Data is valid from 0 to 15.

DP_THRD0 'threshold0'. Data is valid from 0 to 1023.

DP_THRD3 'threshold3'. Data is valid from 0 to 15.

DP_THRD1 ‘threshold1’. Data is valid from 0 to 1023.

CAM+0558h NR1 Defective Pixel Configuration Register 2 NR1_DP2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**DP_THRD4**									
Type							R/W									
Reset							96									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DP_THRD5**									
Type							R/W									
Reset							928									

DP_THRD4 ‘threshold4’. Data is valid from 0 to 1023.

DP_THRD5 ‘threshold5’. Data is valid from 0 to 1023.

Note: DP_THRD5 should be larger than threshold4.

CAM+055Ch NR1 Defective Pixel Configuration Register 3 NR1_DP3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**DP_THRD6**									
Type							R/W									
Reset							48									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DP_THRD7**									
Type							R/W									
Reset							80									

DP_THRD6 ‘threshold6’. Data is valid from 0 to 1023.

DP_THRD7 ‘threshold7’. Data is valid from 0 to 1023.

CAM+0560h NR1 Defective Pixel Configuration Register 4 NR1_DP4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DP_NUM**			**DP_SEL**			**DP_CD7**	**DP_CD6**	**DP_CD5**	**DP_CD4**	**DP_CD3**	**DP_CD2**	**DP_CD1**
Type				R/W			R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset				2			1			0	0	0	0	0	0	0

DP_NUM the max number of defective pixel in the 5x5 window (2 is the recommended value.)

DP_SEL choose which pixel value to compensate.

DP_CD1-7 Enable bits of the conditions for defective pixel detection. When all the enabled conditions are met, the pixel is considered as the defective one.

- **0** Disable
- **1** Enable

CAM+0564h NR1 Crosstalk Compensation Configuration Register NR1_CT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**CT_DIV**	**CT_THRD**							
Type								R/W	R/W							
Reset								1	128							

CT_DIV Selection of the CT compensation value when CT detection is off

- **0** Compensation value is clipped to zero.
- **1** Compensation value is half of the value.

CT_THRD Crosstalk threshold.

CAM+0568h NR1 Noise Reduction Configuration Register 1 NR1_NR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**GNF**			
Type													R/W			
Reset													4			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**S1**			**S2**			**MBND**									
Type	R/W			R/W			R/W									
Reset	2			4			160									

The register is for global configuration of noise reduction

GNF $Gain_x$ value. Data is valid from 0 to 8. The normalized gain is GNF/8.

S1 Weighting value S1. Data is valid from 0 to 4. The normalized gain is S1/4.

S2 Weighting value S2. Data is valid from 0 to 4. The normalized gain is S2/4.

MBND threshold value

CAM+056Ch NR1 Noise Reduction Configuration Register 2 NR1_NR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**GN1**				**GN2**				**GN3**			
Type					R/W				R/W				R/W			
Reset					6				8				10			

The register is for global configuration of noise reduction.

GN1, GN2, GN3 Gain value 1, 2, 3. The data is valid from 5 to 15. The normalized gain is GN/4.

Note: There is a constraint for these parameters. GN3 > GN2 > GN1 > 4

CAM+0570h NR1 Noise Reduction Configuration Register 3 NR1_NR3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**VLR1**								**VLR2**					
Type			R/W								R/W					
Reset			4								4					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**VLR3**								**VLR4**					
Type			R/W								R/W					

© 2009 MediaTek Inc.

Reset			4								4					

VLR1, VLR2, VLR3, VLR4 g(0), g(32), g(64), g(96) values of R channel.

CAM+0574h NR1 Noise Reduction Configuration Register 4 NR1_NR4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**VLR5**								**VLR6**					
Type			R/W								R/W					
Reset			6								6					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**VLR7**								**VLR8**					
Type			R/W								R/W					
Reset			8								8					

VLR5, VLR6, VLR7, VLR8 g(128), g(160), g(192), g(224) values of R channel.

CAM+0578h NR1 Noise Reduction Configuration Register 5 NR1_NR5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**VLGR1**								**VLGR2**					
Type			R/W								R/W					
Reset			2								2					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**VLGR3**								**VLGR4**					
Type			R/W								R/W					
Reset			4								4					

VLGR1, VLGR2, VLGR3, VLGR4 g(0), g(32), g(64), g(96) values of Gr channel.

CAM+057Ch NR1 Noise Reduction Configuration Register 6 NR1_NR6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**VLGR5**								**VLGR6**					
Type			R/W								R/W					
Reset			4								4					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**VLGR7**								**VLGR8**					
Type			R/W								R/W					
Reset			6								6					

VLGR5, VLGR6, VLGR7, VLGR8 g(128), g(160), g(192), g(224) values of Gr channel.

CAM+0580h NR1 Noise Reduction Configuration Register 7 NR1_NR7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**VLGB1**								**VLGB2**					
Type			R/W								R/W					
Reset			2								2					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**VLGB3**								**VLGB4**					
Type			R/W								R/W					
Reset			4								4					

VLGB1, VLGB2, VLGB3, VLGB4 g(0), g(32), g(64), g(96) values of Gb channel.

© 2009 MediaTek Inc.

CAM+0584h NR1 Noise Reduction Configuration Register 8 NR1_NR8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			VLGB5								VLGB6					
Type			R/W								R/W					
Reset			4								4					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VLGB7								VLGB8					
Type			R/W								R/W					
Reset			6								6					

VLGB5, VLGB6, VLGB7, VLGB8 g(128), g(160), g(192), g(224) values of Gb channel.

CAM+0588h NR1 Noise Reduction Configuration Register 9 NR1_NR9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			VLB1								VLB2					
Type			R/W								R/W					
Reset			4								4					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VLB3								VLB4					
Type			R/W								R/W					
Reset			4								4					

VLB1, VLB2, VLB3, VLB4 g(0), g(32), g(64), g(96) values of B channel.

CAM+058Ch NR1 Noise Reduction Configuration Register 10 NR1_NR10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			VLB5								VLB6					
Type			R/W								R/W					
Reset			6								6					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VLB7								VLB8					
Type			R/W								R/W					
Reset			8								8					

VLB5, VLB6, VLB7, VLB8 g(128), g(160), g(192), g(224) values of B channel.

CAM+0600h YCCGO Control Register YCCGO_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ENY1	ENY2	ENY3	ENC1	ENC2	ENC3
Type											R/W	R/W	R/W	R/W	R/W	R/W
Reset											0	0	0	0	0	0

ENY1 Enable bit of luma-knee effect.

- **0** Disable
- **1** Enable

ENY2 Enable bit of luma-inversion effect

- **0** Disable

1 Enable

ENY3 Enable bit of brightness and contrast adjustments

0 Disable

1 Enable

ENC1 Enable bit of manual chroma setting

0 Disable

1 Enable

ENC2 Enable bit of hue adjustment

0 Disable

1 Enable

ENC3 Enable bit of saturation adjustment

0 Disable

1 Enable

Note 1: Only one of ENY1, ENY2, and ENY3 can be '1'.

Note 2: If ENC1 is enabled, ENC2 and ENC3 should be disabled.

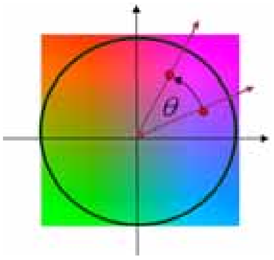
CAM+0604h YCCGO Configuration Register 1 YCCGO_CFG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MU								MV							
Type	R/W								R/W							
Reset	128								128							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	H11								H12							
Type	R/W								R/W							
Reset	127								1							

MU, MV Cb and Cr values when ENC1 is enabled.

H11, H12 YCC_H11 and YCC_H12 in the following figure (represented by the format of 2's complement)

Note: MU and MV is valid from 0 to 255.



$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} \text{YCC_H11} & -\text{YCC_H12} \\ \text{YCC_H12} & \text{YCC_H11} \end{bmatrix} \cdot \begin{bmatrix} Cb - 128 \\ Cr - 128 \end{bmatrix}$$

$$\text{YCC_H11} = 127 \cdot \cos\theta$$

$$\text{YCC_H12} = 127 \cdot \sin\theta$$

CAM+0608h YCCGO Configuration Register 2 YCCGO_CFG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y1								Y2							
Type	R/W								R/W							
Reset	16								32							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y3								Y4							
Type	R/W								R/W							

© 2009 MediaTek Inc.

Reset	224	240

Y1, Y2, Y3, Y4 YCC_Y1, YCC_Y2, YCC_Y3, and YCC_Y4 in the following figure

CAM+060Ch YCCGO Configuration Register 3 YCCGO_CFG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G1								G2							
Type	R/W								R/W							
Reset	64								64							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G3								G4							
Type	R/W								R/W							
Reset	64								64							

G1, G2, G3, G4 YCC_G1, YCC_G2, YCC_G3, and YCC_G4 in the above figure

CAM+0610h YCCGO Configuration Register 4 YCCGO_CFG4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G5								OFSTY							
Type	R/W								R/W							
Reset	64								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OFSTU								OFSTV							
Type	R/W								R/W							
Reset	0								0							

G5 YCC_G5 in the above figure

OFSTY YCC_OFSTY in the following figure (represented by the format of 2's complement)

OFSTU, OFSTV YCC_OFSTU and YCC_OFSTV in the above figure (represented by the format of 2's complement)

Note: YCC_G1 <= YCC_G2 <= YCC_G3; YCC_G5 <= YCC_G4 <= YCC_G3

CAM+0614h YCCGO Configuration Register 5 YCCGO_CFG5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YBNDH								YBNDL							
Type	R/W								R/W							
Reset	255								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									GAINY							
Type									R/W							
Reset									64							

YBNDH The upper clipping value of Y

YBNDL The lower clipping value of Y

GAINY YCC_GAINY in the above figure

CAM+0618h YCCGO Configuration Register 6 YCCGO_CFG6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	UBNDH								UBNDL							
Type	R/W								R/W							

Reset	255								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VBNDH**								**VBNDL**							
Type	R/W								R/W							
Reset	255								0							

UBNDH The upper clipping value of U
UBNDL The lower clipping value of U
VBNDH The upper clipping value of V
VBNDL The lower clipping value of V

CAM+1000h AE Window Result 1 AEMEM0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AESUM1[7:0]**								**AESUM0[23:16]**							
Type	R								R							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AESUM0[15:0]**															
Type	R															
Reset																

AESUM0 AE window 0 summation
AESUM1 AE window 1 summation low byte

CAM+1004h AE Window Result 2 AEMEM1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AESUM2[15:0]**															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AESUM1[23:8]**															
Type	R															
Reset																

AESUM1 AE window 1 summation high bytes
AESUM2 AE window 2 summation low bytes

CAM+1008h AE Window Result 3 AEMEM2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**AESUM3[23:8]**															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AESUM3[7:0]**								**AESUM2[23:16]**							
Type	R								R							
Reset																

AESUM3 AE window 0 summation
AESUM2 AE window 1 summation high byte

 © 2009 MediaTek Inc.

CAM+100Ch AE Window Result 4 AEMEM3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE								AESUM4[23:16]							
Type	R								R							
Reset	0xae															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM4[15:0]															
Type	R															
Reset																

AESUM4 AE window 4 summation

CAM+1010h AE Window Result 5 AEMEM4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM6[7:0]								AESUM5[23:16]							
Type	R								R							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM5[15:0]															
Type	R															
Reset																

AESUM5 AE window 5 summation

AESUM6 AE window 6 summation low byte

CAM+1014h AE Window Result 6 AEMEM5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM7[15:0]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM6[23:8]															
Type	R															
Reset																

AESUM6 AE window 6 summation high bytes

AESUM7 AE window 7 summation low bytes

CAM+1018h AE Window Result 7 AEMEM6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM8[23:8]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM8[7:0]								AESUM7[23:16]							
Type	R								R							
Reset																

AESUM8 AE window 0 summation

AESUM7 AE window 1 summation high byte

CAM+101Ch AE Window Result 8 AEMEM7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE								AESUM9[23:16]							
Type	R								R							
Reset	0xae															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM9[15:0]															
Type	R															
Reset																

AESUM9 AE window 9 summation

CAM+1020h AE Window Result 9 AEMEM8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM11[7:0]								AESUM10[23:16]							
Type	R								R							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM10[15:0]															
Type	R															
Reset																

AESUM10 AE window 10 summation

AESUM11 AE window 11 summation low byte

CAM+1024h AE Window Result 10 AEMEM9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM12[15:0]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM11[23:8]															
Type	R															
Reset																

AESUM11 AE window 11 summation high bytes

AESUM12 AE window 12 summation low bytes

CAM+1028h AE Window Result 11 AEMEM10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM13[23:8]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM13[7:0]								AESUM12[23:16]							
Type	R								R							
Reset																

AESUM12 AE window 12 summation

AESUM13 AE window 13 summation high byte

© 2009 MediaTek Inc.

CAM+102Ch AE Window Result 12 AEMEM11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE								AESUM14[23:16]							
Type	R								R							
Reset	0xae															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM14[15:0]															
Type	R															
Reset																

AESUM14 AE window 9 summation

CAM+1030h AE Window Result 13 AEMEM12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM16[7:0]								AESUM15[23:16]							
Type	R								R							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM15[15:0]															
Type	R															
Reset																

AESUM15 AE window 10 summation

AESUM16 AE window 11 summation low byte

CAM+1034h AE Window Result 14 AEMEM13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM17[15:0]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM16[23:8]															
Type	R															
Reset																

AESUM16 AE window 16 summation high bytes

AESUM17 AE window 17 summation low bytes

CAM+1038h AE Window Result 15 AEMEM14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM18[23:8]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM18[7:0]								AESUM17[23:16]							
Type	R								R							
Reset																

AESUM17 AE window 17 summation high byte

AESUM18 AE window 18 summation

 © 2009 MediaTek Inc.

CAM+103Ch AE Window Result 16 AEMEM15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE								AESUM19[23:16]							
Type	R								R							
Reset	0xae															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM19[15:0]															
Type	R															
Reset																

AESUM19 AE window 19 summation

CAM+1040h AE Window Result 17 AEMEM16

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM21[7:0]								AESUM20[23:16]							
Type	R								R							
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM20[15:0]															
Type	R															
Reset																

AESUM20 AE window 20 summation

AESUM21 AE window 21 summation low byte

CAM+1044h AE Window Result 18 AEMEM17

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM22[15:0]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM21[23:8]															
Type	R															
Reset																

AESUM21 AE window 21 summation high bytes

AESUM22 AE window 22 summation low bytes

CAM+1048h AE Window Result 19 AEMEM18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AESUM23[23:8]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM23[7:0]								AESUM22[23:16]							
Type	R								R							
Reset																

AESUM22 AE window 22 summation high byte

AESUM23 AE window 23 summation

© 2009 MediaTek Inc.

CAM+104Ch AE Window Result 20 AEMEM19

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AE								AESUM24[23:16]							
Type	R								R							
Reset	0xae															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AESUM24[15:0]															
Type	R															
Reset																

AESUM24 AE window 24 summation

CAM+(1050~1074)h Flare Histogram (1 ~ 10) FLAREMEM(0~9)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											FLAREB(0~9)[21:16]					
Type	R										R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FLAREB(0~9)[15:0]															
Type	R															
Reset																

FLAREB(0~9) Flare histogram bin (0~9)

CAM+(1078~1174)h AE Histogram (1 ~ 64) AEHIS(0~63)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													AEHIS(0~63)[19:16]			
Type	R												R			
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AEHIS(0~63)[15:0]															
Type	R															
Reset																

AEHIS(0~63) AE histogram bin (0~63)

CAM+(1178~1234)h AWB XY Window Result (1 ~ 12) (Count) AWBXY_RLT(0~12)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBOUT_WCOUNT(0~11)[19:16]					
Type	R										R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_WCOUNT(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_WCOUNT(0~11) AWB XY Window Count

CAM+(1178~1234)h AWB XY Window Result (1 ~ 12) (Rsum) AWBXY_RLT(0~12)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_WRSUM(0~11)[31:16]															
Type	R															

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_WRSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_WRSUM(0~11) AWB XY Window R Sum

CAM+(1178~1234)h AWB XY Window Result (1 ~ 12) (Gsum) AWBXY_RLT(0~12)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_WGSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_WGSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_WGSUM(0~11) AWB XY Window G Sum

CAM+(1178~1234)h AWB XY Window Result (1 ~ 12) (Bsum) AWBXY_RLT(0~12)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_WBSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_WBSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_WBSUM(0~11) AWB XY Window B Sum

CAM+1238h AWB Sum Window Result (1 ~ 12) (Count) AWBSUM_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBOUT_FCOUNT[19:16]					
Type	R										R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_FCOUNT[15:0]															
Type	R															
Reset																

AWBOUT_FCOUNT AWB Sum Window Count

CAM+123Ch AWB Sum Window Result (Rsum) AWBSUM_RSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_FRSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_FRSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_FRSUM	AWB Sum Window R Sum

CAM+1240h AWB Sum Window Result (Gsum) AWBSUM_GSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_FGSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_FGSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_FGSUM	AWB Sum Window G Sum

CAM+1244h AWB Sum Window Result (Bsum) AWBSUM_BSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_FBSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_FBSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_FBSUM	AWB Sum Window B Sum

CAM+1248h AWB Color Edge Window Result (1 ~ 12) (Count) AWBCE_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											AWBOUT_CECOUNT[19:16]					
Type	R										R					
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_CECOUNT[15:0]															
Type	R															
Reset																

AWBOUT_CECOUNT	AWB Color Edge Window Count

CAM+124Ch AWB Color Edge Window Result (Rsum) AWBCE_RSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_CERSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_CERSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_CERSUM	AWB Color Edge Window R Sum

CAM+1250h AWB Color Edge Window Result (Gsum) AWBCE_GSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	AWBOUT_CEGSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_CEGSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_CEGSUM AWB Color Edge Window G Sum

CAM+1254h AWB Color Edge Window Result (Bsum) AWBCE_BSUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AWBOUT_CEBSUM(0~11)[31:16]															
Type	R															
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AWBOUT_CEBSUM(0~11)[15:0]															
Type	R															
Reset																

AWBOUT_CEBSUM AWB Color Edge Window B Sum

CAM+6000 - 6100h HST BIN01 - 64 HST_BIN01 - 64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BIN01 – 64															
Type	R															
Reset	0															

The registers are used to read the histogram from bin01 to bin64, which is used by GDC
Note: The enable bit of HST function (HST_CON/CAM+0x540) should be set as low ‘0’, while reading the values

5.6 Capture Resize

5.6.1 General Description

This block provides the image resizing function for image and video capturing scenarios. It receives image data from the ISP module, performs the image resizing function and outputs to the IMG_DMA module. **Figure 116** shows the block diagram. The capture resize is composed of horizontal and vertical resizing blocks. It can scale up or down the input image by any ratio. However, the maximum sizes of input and output images are limited to 4095x4095 with limitation specified on chapter 1.1.3 application notes.

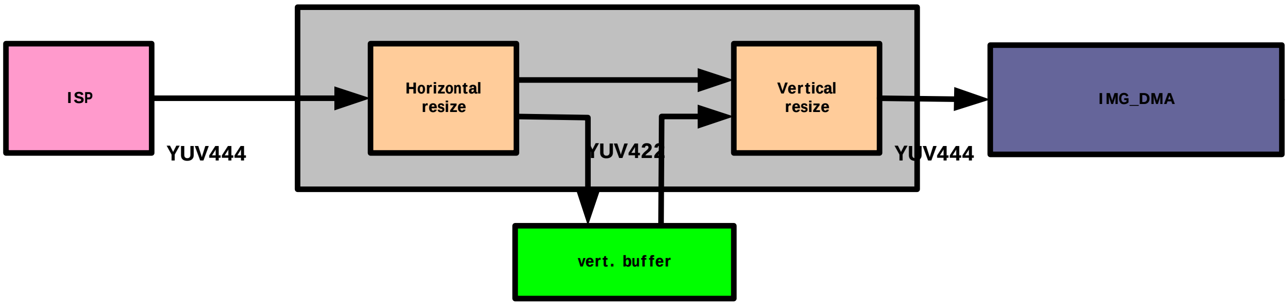


Figure 116 ***Block diagram of the capture resize***

The main algorithm of resizing function is cubic interpolation. The input and output format are both YUV444. But the internal working memory format is YUV422 to mitigate memory and bandwidth requirements.

5.6.2 Register Definitions

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CRZ+ 0000h	Capture Resize Configuration Register	CRZ_CFG
CRZ + 0004h	Capture Resize Control Register	CRZ_CON
CRZ + 0008h	Capture Resize Status Register	CRZ_STA
CRZ + 000Ch	Capture Resize Interrupt Register	CRZ_INT
CRZ + 0010h	Capture Resize Source Image Size Register 1	CRZ_SRCSZ1
CRZ + 0014h	Capture Resize Target Image Size Register 1	CRZ_TARSZ1
CRZ + 0018h	Capture Resize Horizontal Ratio Register 1	CRZ_HRATIO1
CRZ + 001Ch	Capture Resize Vertical Ratio Register 1	CRZ_VRATIO1
CRZ + 0040h	Capture Resize Coefficient Table	CRZ_FRCFG

5.6.2.1 Capture Resize Configuration Register

CRZ+0000h Capture Resize Configuration Register CRZ_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							LBMAX									
Type							R/W									
Reset							0									
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		NORFDB	VSRSTEN	ECV	INTEN[1]	INTEN[0]	LBSEL	PCON		O_Y2R1	O_IPP1	O_OVL		SRC		
Type		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W		R/W		
Reset		0	0	0	0	0	1	0		0	0	0		0		

The register is for global configuration of Capture Resize.

SRC The register field specifies which source is serviced.

- **0** Camera Interface
- **1** Image Rotator 0
- **2** MPEG4 Deblocking

3 PRZ
4 R2Y 0
Others Reserved

O_OVL The register field determines output target
0 Disable output to Overlay
1 Enable output to Overlay

O_IPP1 The register field determines output target
0 Disable output to IPP 1
1 Enable output to IPP 1

O_Y2R1 The register field determines output target
0 Disable output to Y2R 1
1 Enable output to Y2R 1

PCON The register bit specifies whether resizing continues or not whenever an image finishes processing. If to stop immediately is desired, reset Capture Resize directly. If the last image is desired, set the register bit to ‘0’ first. Then wait until image resizer is not busy again. Finally reset image resizer.
0 Single run
1 Continuous run

LBSEL Line buffer selection.
0 CRZ cant work!
1 Use **RESZ_LB** (resizer dedicated line buffer) as temporary buffer.

INTEN[0] Frame End Interrupt Enable. When interrupt is enabled, an interrupt is generated whenever CRZ finishes.
0 Interrupt for frame done is disabled.
1 Interrupt for frame done is enabled.

INTEN[1] Frame Start Interrupt Enable. When interrupt is enabled, an interrupt is generated whenever CRZ receives Vsync signal from ISP.
0 Interrupt for frame start is disabled.
1 Interrupt for frame start is enabled.

ECV The register field determines whether using ‘ec’ algorithm for vertical downscaling
0 Use cubic algorithm
1 Use ec algorithm. It helps when bandwidth is very critical.

VSRSTEN The register field determines whether force CRZ reset when vsync arise but previous frame not done yet.
0 Not force reset
1 Force reset when vsync

NORFDB The register field determines not double buffer some registers.
0 Double buffering registers
1 No double buffering registers

LBMAX **Number of lines used in upsampling scenario :**

```
WMIN = ((WS > WT) ? WT : WS);  // use for Width down, and Height up
WMIN_EVEN = WMIN + WMIN%2;
LB = (int)(2688 / WMIN_EVEN) * 6;
LBMAX = LB > 1023 ? 1023 : LB;
```

5.6.2.2 Capture Resize Control Register

CRZ+0004h Capture Resize Control Register CRZ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**RST**
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Name																**ENA**
Type																R/W
Reset																0

The register is for global control of Capture Resize. **Note that software reset does NOT reset all register settings. Remember to trigger Capture Resize first before triggering image sources to Capture Resize.**

ENA Writing ‘1’ to the register bit causes CRZ proceed to work.

RST Writing ‘1’ to the register causes CRZ to stop immediately and keep in reset state. In order to go to normal state, write ‘0’ to the register bit.

5.6.2.3 Capture Resize Status Register

CRZ+0008h Capture Resize Status Register CRZ_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**BUSYI**	**BUSYO**
Type															RO	RO
Reset																

The register indicates global status of Capture Resize.

BUSYI Input interface busy.

BUSYO Output interface busy.

5.6.2.4 Capture Resize Interrupt Register

CRZ+000Ch Capture Resize Interrupt Register CRZ_INT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**FSTINT**	**FEDINT**
Type															RC	RC
Reset															0	0

The register shows up the interrupt status of resizer.

 © 2009 MediaTek Inc.

FEDINT Interrupt for CRZ. No matter the register bit CRZ_CFG.INTEN[0] is enabled or not, the register bit is active whenever CRZ completes. It could be as software interrupt by polling the register bit. Clear it by reading the register.

FSTINT Interrupt for CRZ. No matter the register bit CRZ_CFG.INTEN[1] is enabled or not, the register bit is active whenever CRZ start working. It could be as software interrupt by polling the register bit. Clear it by reading the register.

5.6.2.5 Capture Resize Source Image Size Register 1

CRZ+0010h Capture Resize Source Image Size Register 1 CRZ_SRCSZ1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					HS											
Type					R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					WS											
Type					R/W											

The register specifies the size of source image after coarse shrink process. **The allowable maximum size is 4095x4095 with limitation written in application notes.**

WS The register field specifies the width of source image after coarse shrink process.

- **1** The width of source image after coarse shrink process is 1.
- **2** The width of source image is 2.
- …

HS The register field specifies the height of source image after coarse shrink process.

- **1** The height of source image after coarse shrink process is 1.
- **2** The height of source image after coarse shrink process is 2.
- …

5.6.2.6 Capture Resize Target Image Size Register 1

CRZ+0014h Capture Resize Target Image Size Register 1 CRZ_TARSZ1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					HT											
Type					R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					WT											
Type					R/W											

The register specifies the size of target image. **The allowable maximum size is 4095x4095 with limitation written in application notes.**

WT The register field specifies the width of target image.

- **1** The width of target image is 1.
- **2** The width of target image is 2.
- …

HT The register field specifies the height of target image.

- **1** The height of target image is 1.
- **2** The height of target image is 2.

…

5.6.2.7 Capture Resize Horizontal Ratio Register 1

CRZ+0018h Capture Resize Horizontal Ratio Register CRZ_HRATIO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RATIO [31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RATIO [15:0]**															
Type	R/W															

The register specifies horizontal resizing ratio. It is obtained by

CRZ_SRCSZ.HS > CRZ_TARSZ.HT

? ((CRZ_TARSZ.HT - 1)* 2^{20} + (CRZ_SRCSZ.HS -1)/2) / (CRZ_SRCSZ.HS -1)

: ((CRZ_SRCSZ.HS - 1)* 2^{20} + (CRZ_TARSZ.HT-1)/2) / (CRZ_TARSZ.HT-1).

5.6.2.8 Capture Resize Vertical Ratio Register 1

CRZ+001Ch Capture Resize Vertical Ratio Register 1 CRZ_VRATIO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RATIO [31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RATIO [15:0]**															
Type	R/W															

The register specifies vertical resizing ratio. It is obtained by

CRZ_SRCSZ.VS > CRZ_TARSZ.VT

? ((CRZ_TARSZ.VT - 1)* 2^{20} + (CRZ_SRCSZ.VS -1)/2) / (CRZ_SRCSZ.VS -1)

: ((CRZ_SRCSZ.VS - 1)* 2^{20} + (CRZ_TARSZ.VT-1)/2) / (CRZ_TARSZ.VT-1).

5.6.2.9 Capture Resize Coefficient Table Register

CRZ+0040h Capture Resize Coefficient Table CRZ_FRCFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**USEL**								**DSEL**				
Type				R/W								R/W				
Reset				0	1	1	0	0				0	1	1	0	0

The register specifies the coefficient table for resizing. Valid number is form 0 to 19. While '1' is the most blur and '19' is the sharpest. '0' is a special case, linear interpolation rather than cubic interpolation is used.

USEL choose 'USEL' > 12 may get undesirable result, '8' is recommended.

DSEL '15' is recommended.

5.6.2.10 Capture Resize Busy Register

CRZ+0044h Capture Resize Busy CRZ_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DTH								UTH							
Type	R/W								R/W							
Reset	0								0							

The register specifies when to issues a busy signal to inform image dma that the buffer in crz is nearly full.

UTH Threshold setting for upsampling

- **0xff** crz_busy when used line buffer >=6
- **0xfe** crz_busy when used line buffer >=8
- **0xfc** crz_busy when used line buffer >=16
- **0xf8** crz_busy when used line buffer >=32
- **0xf0** crz_busy when used line buffer >=64
- **0xe0** crz_busy when used line buffer >=128
- **0xc0** crz_busy when used line buffer >=256
- **0x80** crz_busy when used line buffer >=512
- **0x00** crz_busy always low

Recommended: 0xff or half of max line buffer

DTH Threshold setting for downsampling

- **Bit 7** crz_busy when write pointer and read pointer are not at the same line buffer
- **Bit 6** crz_busy when write pointer bit 10 = 1
- **Bit 5** crz_busy when write pointer bit 9 = 1
- **Bit 4** crz_busy when write pointer bit 8 = 1
- **Bit 3** crz_busy when write pointer bit 7 = 1
- **Bit 2** crz_busy when write pointer bit 6 = 1
- **Bit 1** crz_busy when write pointer bit 5 = 1
- **Bit 0** crz_busy when write pointer bit 4 = 1

Recommended: 0x80 or 0xc0

5.6.3 Application Notes

- SRCSZ and TARSZ limitation for upscaling

  3 <= CRZ_SRCSZ.WS <= 2688;
  3 <= CRZ_SRCSZ.HS <= 2688;
  (CRZ_TARSZ.WT-1) / (CRZ_SRCSZ.WS-1) <= 64;

- SRCSZ and TARSZ limitation for downcaling

  2 <= CRZ_TARSZ.WT <= 2688;
  2 <= CRZ_TARSZ.HT <= 2688;
  (CRZ_TARSZ.WT-1) / (CRZ_SRCSZ.WS-1) >= 1/2048;

- Configuration procedure for pixel-based image sources

CRZ_SRCSZ = source image size;
CRZ_TARSZ = target image size;
CRZ_CFG.LBSEL = 1; // must be 1, crz cant work when set to 0
CRZ_CFG.SRC = **0~4**;
CRZ_CFG.O_OVL = **1 or 0;**
CRZ_CFG.O_IPP1 = **1 or 0;**
CRZ_CFG.O_Y2R1 = **1 or 0;**
WMIN = (WS > WT) ? WT : WS; // use for Width down and Height up
WMIN_EVEN = WMIN + WMIN%2;
LB = (int)(2688/ WMIN_EVEN) * 6;
CRZ_CFG.LBMAX = (LB > 1023) ? 1023 : LB;
CRZ_HRATIO = horizontal ratio;
CRZ_VRATIO = vertical ratio;
CRZ_CON = 0x1;

5.7 CEVASYS Subsystem

The CEVASYS subsystem consists of three modules, namely cevaX1620, cevaxs1200_sys, and ceva_ap. The cevaX1620 itself is a high performance DSP. The cevaxs1200_sys is the circuit surrounding the cevaX1620. The ceva_ap is the interface between cevaxs1200_sys and the system. Some important operating information for CEVASYS is under the control of ARM9. Here defines these control registers.

5.7.1 CEVASYS AP configure registers

Register Address	Register Name	Synonym
CEVA_CFG + 0000h	CEVA DSP Control Register 0	CEVA_CON0
CEVA_CFG + 0004h	CEVA DSP Control Register 1	CEVA_CON1
CEVA_CFG + 0008h	CEVA CG_CON Register	CEVA_CG_CON
CEVA_CFG + 000Ch	CEVA External Memory Offset	CEVA_EMO
CEVA_CFG + 0010h	CEVA DELSEL Control Register 0	CEVA_DELSEL0
CEVA_CFG + 0014h	CEVA DELSEL Control Register 1	CEVA_DELSEL1
CEVA_CFG + 0018h	CEVA DELSEL Control Register 2	CEVA_DELSEL2
CEVA_CFG + 0000h	CEVA DSP Control Register 0	CEVA_CON0

Table 99 ***CEVASYS AP Configure Registers***

5.7.2 Register Definition

CEVA_CFG +0000h **CEVA Control 0 Register** **CEVA_CON0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAKE**							**CXRSTN**				**GLOBRSTN**				**BOOT**

Type	R/W	R/W	R/W	R/W				R/W				R/W				R/W
Reset	0x0							0				0				0

CEVA control 0 register.

BOOT The cevaX1620 booting selection bit.

- **0** The cevaX1620 boots from address 0x0000.
- **1** Takes EXT_VECTOR[31:0], defined in CEVA_CON1 register, as the booting address.

GLOBRSTN The global reset bit of cevaxs1200, including cevaX1620 and cevaxs1200_sys.

CXRSTN The reset bit of cevaX1620.

WAKE The WAKE signals.

CEVA_CFG +0004h CEVA Control 1 Register CEVA_CON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EXT_VECTOR[31:16]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXT_VECTOR[15:0]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The booting vector when the BOOT bit is set.

CEVA_CFG +0008h CEVA CG_CON Register CEVA_CG_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**CCIF_CG_CON**				**AHB_CG_CON**
Type												R/W				R/W
Reset												1				1

The CEVASYS clock gating control bits.

AHB_CG_CON The clock gating control bit of the AHB clock.

- **0** Release the clock.
- **1** Stop the clock.

CCIF_CG_CON The clock gating control bit of the AHB clock for the CCIF module

- **0** Release the clock.
- **1** Stop the clock.

CEVA_CFG +000Ch CEVA External Memory Offset CEVA_EMO

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CEVA_EMO[15:0]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0xC000															

The external memory offset register for cevaxs1200. It defines how the address should be translated from CEVASYS domain to the system bus.

CEVA_CFG +0010h CEVA DELSEL Control Register 0 CEVA_DELSEL 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CEVA_DELSEL0[31:16]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0xAAAA															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CEVA_DELSEL0[15:0]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0xAAAA															

CEVA_DELSEL0 This register is related to the speed configuration of SRAM.

CEVA_CFG +0014h CEVA DELSEL Control Register 1 CEVA_DELSEL 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CEVA_DELSEL1[31:16]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0xAAAA															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CEVA_DELSEL1[15:0]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0xAAAA															

CEVA_DELSEL1 This register is related to the speed configuration of SRAM.

CEVA_CFG +0018h CEVA DELSEL Control Register 2 CEVA_DELSEL 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**CEVA_DELSEL2[11:0]**											

Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0x6FF											

CEVA_DELSEL2 This register is related to the speed configuration of SRAM.

5.8 Display Pixel Interface Controller

5.8.1 Features

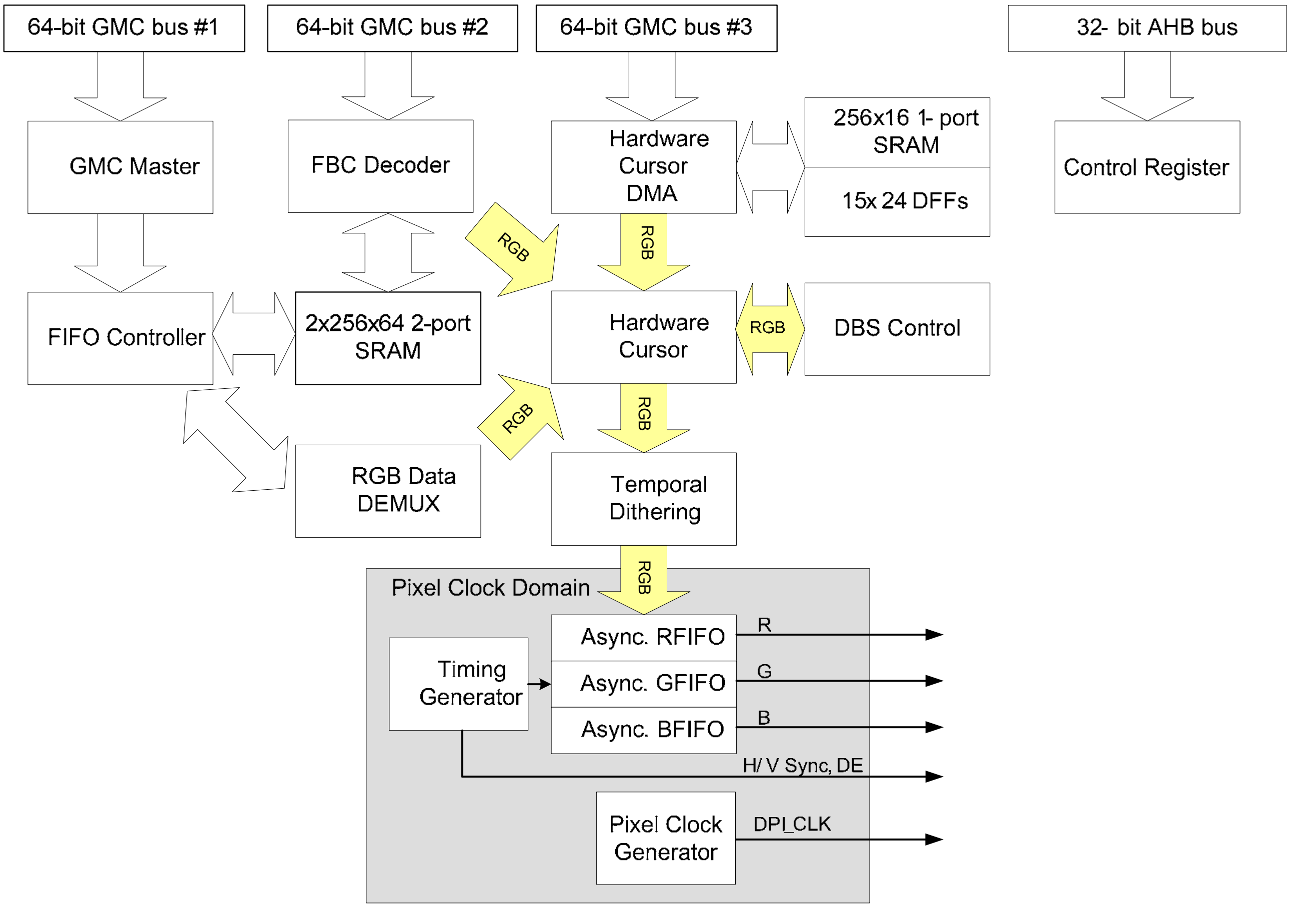


Figure 117 ***The block diagram of display pixel interface controller***

Figure 12 gives the block diagram of display pixel interface (abbreviated to DPI) controller. The DPI controller generates appropriate timing signals for generic LCD module with digital RGB interface and moves the display data from the frame buffer to the interface. It has the following features:

1. Supports LCD panel with resolution up to 1024x1024 (frame buffer in off-chip SDRAM)
2. Supports RGB888 and RGB565 color mode
3. Supports block reading (for hardware scrolling)
4. Supports one 32x32 or four 16x16 4 bit indexed hardware cursor with transparent color key
5. Supports 2X or 4X temporal dithering for 6-bit LCD panel

5.8.2 Control Registers

5.8.2.1 Register Definition

+0000h DPI Control Register DPI_CNTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DPI_OEN_OFF	INTF_RGB_ORDER	DSI_MODE	FBC_EN	DBS_EN	INTF_68_EN	SRC_RGB_ORDER	PIXEL_FMT	FB_CHKEN	ADP_FIFO_EN	FAST_RC_EN	FS_EN		FB2_ENABLE	FB1_ENABLE	DPI_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0

DPI_EN Turn the DPI controller on/off
- **0** DPI controller off
- **1** DPI controller on

FB1_ENABLE Enable/disable external frame buffer 1
- **0** Off
- **1** On

FB2_ENABLE Enable/disable external frame buffer 2
- **0** Off
- **1** On

FS_EN Frame synchronization enable. When more than 1023 pixels are lost, this function performs synchronization between frame data and DPI protocol.
- **0** Off
- **1** On

FAST_RC_EN Fast recovery enable. This function speeds up frame synchronization by dropping frame data to find the starting of the frame. FS_EN should also be enabled for this function to work.
- **0** Off
- **1** On

ADP_FIFO_EN Adaptive FIFO high/low threshold control. FIFO high/low threshold would automatically increase by FIFO_TH_INC when GMC FIFO is empty.
- **0** Off
- **1** On

FB_CHKEN Keep reading current frame buffer if next frame buffer is not completely prepared by LCD.
- **0** Disable
- **1** Enable

PIXEL_FMT Specify the pixel color format
- **0** RGB565
- **1** RGB888

SRC_RGB_ORDER Specify the source RGB color order (From MSB to LSB)
- **0** RGB

1 BGR

INTF_68_EN Sequential 6-/8-bit output interface enable

0 Off

1 On

DBS_EN Display backlight scaling path enable. DPI_CUSR_CNTL.CURSOR_EN shall be 0 if DBS_EN is on.

0 Off

1 On

FBC_EN Frame buffer decompression path enable. DPI_CUSR_CNTL.CURSOR_EN and DPI_DITHER.DITHER_MODE shall be 0 if FBC_EN is on.

0 Off

1 On

DSI_MODE Display serial interface (DSI) path enable

0 Off

1 On

INTF_RGB_ORDER Output interface RGB order selection

0 RGB

1 BGR

DPI_OEN_OFF Turn off DPI output enable

0 Do not turn off

1 Turn off

+0004h DPI IRQ Enable Register DPI_INTEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						OVERSPEC_EN	FBC_ERR_EN	VSYNC_EN	FRM_ERR_EN	LINE_ERR_EN	CNT_ERR_EN	B_EMPTY_EN	G_EMPTY_EN	R_EMPTY_EN	GMC_FULL_EN	GMC_EMPY_EN
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0

This register sets the interrupt enable of individual interrupt source generated by DPI controller.

GMC_EMPTY_EN Interrupt issues when source data FIFO is empty.

0 Disable

1 Enable

GMC_FULL_EN Interrupt issues when source data FIFO is full.

0 Disable

1 Enable

R_EMPTY_EN Interrupt issues when Red FIFO is empty

0 Disable

1 Enable

G_FIFO_EMPTY_EN Interrupt issues when Green FIFO is empty

- **0** Disable
- **1** Enable

B_FIFO_EMPTY_EN Interrupt issues when Blue FIFO is empty

- **0** Disable
- **1** Enable

CNT_ERR_EN Total lost pixels due to bandwidth shortage exceed 1024, an interrupt request will be generated

- **0** Disable
- **1** Enable

LINE_ERR_EN If there are still lost pixels when HSYNC is activated, an interrupt request will be generated

- **0** Disable
- **1** Enable

FRAME_ERR_EN If there are still lost pixels when VSYNC is activated, an interrupt request will be generated

- **0** Disable
- **1** Enable

VSYNC_EN If VSYNC is activated, an interrupt request will be generated

- **0** Disable
- **1** Enable

FBC_ERR_EN An interrupt issues when frame buffer decompression error.

- **0** Disable
- **1** Enable

OVERSPEC_EN If ADP_FIFO_EN is enabled, an interrupt issues when FIFO_HIGH_TH is larger than or equal to FIFO_TH_MAX.

- **0** Disable
- **1** Enable

+0008h DPI IRQ Status Register DPI_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						OVERPSEC	FBC_ERR	VSYNC	FRAME_ERR	LINE_ERR	CNT_ERR	BFIFO_EMPTY	GFIFO_EMPTY	RFIFO_EMPTY	GMC_FULL	GMC_EMPTY
Type						RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC

This read only register gives the current interrupt request status of the DPI controller. Each field is corresponded to the previous DPI_INTEN register.

 © 2009 MediaTek Inc.

+0010h DPI Size Register DPI_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							V_SIZE									
Type							R/W									
Reset							0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							H_SIZE									
Type							R/W									
Reset							0									

This register sets the horizontal (x-axis) and vertical (y-axis) resolution of the currently used LCD panel. For an LCD panel with resolution **320x240**, the H_SIZE field should be set to (320-1) = **319** and the V_SIZE field should be set to (240-1) **239**.

H_SIZE Set the horizontal pixel number.
V_SIZE Set the vertical pixel number.

+0014h DPI Clock Control Register DPI_CLKCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DPI_CK_POL															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DPI_CK_DUT								DPI_CK_DIV				
Type				R/W								R/W				
Reset				1								1				

DPI_CK_DIV Set the divisor of the pixel PLL clock for DPI clock generation.

- **0** DPI_CLOCK stops
- **1** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 2$
- **2** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 3$
- **3** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 4$
- **4** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 5$
- **5** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 6$
- **6** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 7$
- **7** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 8$
- **…**
- **31** $FREQ_{DPI_CLOCK} = FREQ_{PLL_CLOCK} / 32$

DPI_CK_DUT Set the output DPI clock duty cycle, should be **less than or equal to DPI_CK_DIV**

- **0** DPI_CLOCK stops
- **1~31** Wave form of DPI_CLOCK is {DPI_CK_DUT, DPI_CK_DIV – DPI_CK_DUT + 1}

DPI_CK_POL Set the polarity of the output DPI clock

- **0** The same as the internal DPI clock
- **1** Inverse to the internal DPI clock

+0018h DPI Dither Control Register DPI_DITHER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														R_LBITS_SEL		
Type														R/W		
Reset														000		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		G_LBITS_SEL				B_LBITS_SEL				IMPOSE_SEL					DITHER_MODE	
Type		R/W				R/W				R/W					R/W	
Reset		000				000				010					0	

DITHER_MODE Specify the temporal dithering mode. DPI_CNTL.FBC_EN shall be 0 if DITHER_MODE is not off.

1X 4X mode
01 2X mode
00 off

IMPOSE_SEL Select the intensity of dithering

000 impose the dither effect on bit0
001 impose the dither effect on bit1
010 impose the dither effect on bit2
011 impose the dither effect on bit3
100 impose the dither effect on bit4
101 impose the dither effect on bit5
110 impose the dither effect on bit6
111 impose the dither effect on bit7

B_LBITS_SEL Select the dithering reference bits of blue component

000 bit[1:0]
001 bit[2:1]
010 bit[3:2]
011 bit[4:3]
100 bit[5:4]
101 bit[6:5]
110 bit[7:6]
111 bit[7:6]

G_LBITS_SEL Select the dithering reference bits of green component

000 bit[1:0]
001 bit[2:1]
010 bit[3:2]
011 bit[4:3]
100 bit[5:4]
101 bit[6:5]
110 bit[7:6]
111 bit[7:6]

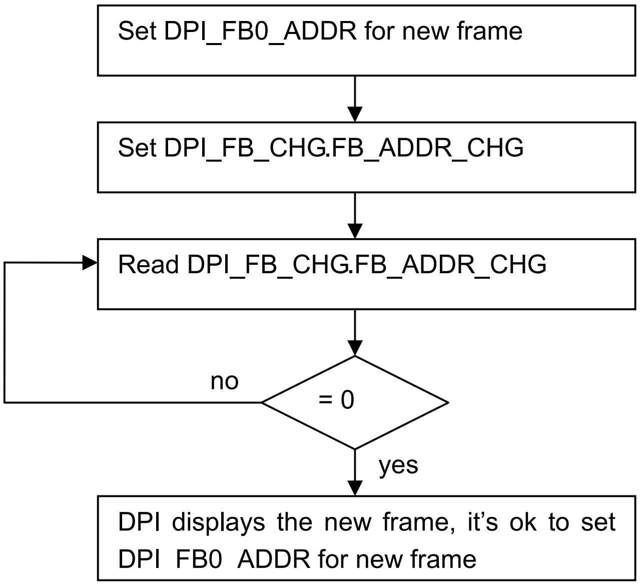
R_LBITS_SEL Select the dithering reference bits of red component

000 bit[1:0]
001 bit[2:1]
010 bit[3:2]
011 bit[4:3]
100 bit[5:4]
101 bit[6:5]
110 bit[7:6]
111 bit[7:6]

+001Ch DPI Frame Buffer Address Change Register DPI_FB_CHG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**FB_ADDR_CHG**
Type																R/W
Reset																0

FB_ADDR_CHG Waiting for Frame buffer address changing. It is used when single frame buffer. When software changes the single frame buffer address (register DPI_FB0_ADDR) for a new frame, FB_ADDR_CHG should also be set. And then after DPI starts to display the new frame, this bit will be clear. Software reads this bit to check if the new frame starts to display. If the value of this bit is 0, it means DPI starts to display the new frame, and it is ok to change DPI_FB0_ADDR for the next frame. If the value of this bit is 1, it means DPI still displays old frame, and doesn't use the new value of DPI_FB0_ADDR. Software can't change DPI_FB0_ADDR for the next frame. Please see Figure 2.



© 2009 MediaTek Inc.

Figure 118 ***Frame buffer address change flow for single frame buffer***

+0020h DPI Frame Buffer 0 Address Register DPI_FB0_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DPI_FB0_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**DPI_FB0_ADDR[15:0]**															
Type	R/W															
Reset	0															

This register gives the display start address of the external frame buffer 0.

+0024h DPI Frame Buffer 0 Line Step Register DPI_FB0_STEP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					**DPI_FB0_STEP**											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0

This register gives the horizontal line length (line incremental step) in byte of the external frame buffer 0.

+0028h DPI Frame Buffer 1 Address Register DPI_FB1_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DPI_FB1_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**DPI_FB1_ADDR[15:0]**															
Type	R/W															
Reset	0															

This register gives the display start address of the external frame buffer 1.

+002ch DPI Frame Buffer 1 Line Step Register DPI_FB1_STEP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					**DPI_FB1_STEP**											
Type					R/W											
Reset					0											

This register gives the horizontal line length (line incremental step) in byte of the external frame buffer 1.

 © 2009 MediaTek Inc.

+0030h DPI Frame Buffer 2 Address Register DPI_FB2_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DPI_FB2_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DPI_FB2_ADDR[15:0]															
Type	R/W															
Reset	0															

This register gives the display start address of the external frame buffer 2.

+0034h DPI Frame Buffer 2 Line Step Register DPI_FB2_STEP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					DPI_FB2_STEP											
Type					R/W											
Reset					0											

This register gives the horizontal line length (line incremental step) in byte of the external frame buffer 2.

+0038h DPI Overlay Engine Control Register DPI_OVL_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							FB_LINE_TH									
Type							R/W									
Reset							0									
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							FB_WORD_TH									
Type							R/W									
Reset							0									

This register control the timing of overlay engine start to update frame buffer once DPI read “FB_LINE_TH” line and “FB_WORD_TH” words(4bytes) from any frame buffer.

+003Ch DPI FBC Line Length Register DPI_FBCD_LINE_L

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					LINE_W											
Type					R/W											
Reset					0											

LINE_W Line length of compressed frames, in unit of byte.

+0040h DPI FIFO Threshold Control Register DPI_ FIFO_TH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							FIFO_HIGH_TH									
Type							R/W									
Reset							0									
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							FIFO_LOW_TH									
Type							R/W									
Reset							0									

DPI internal FIFO high/low thresholds control. When the number of FIFO valid entries is smaller than FIFO_LOW_TH, DPI will request external memory bandwidth with ultra-high priority until the number of FIFO valid entries is greater than or equal to FIFO_HIGH_TH. Note the FIFO threshold is in unit of entry. Each entry is 8 bytes. The maximum value of FIFO threshold is 512. It's suggested that the FIFO_HIGH_TH should not be greater than 496 (512-16) since the number of valid entries may never greater be than 496.

+0044h DPI FIFO Threshold Increment Register DPI_ FIFO_INC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name											TEST_SUM_SEL	FIFO_TH_INC				
Type											R/W	R/W				
Reset											0	0				

FIFO_TH_INC DPI internal FIFO high/low thresholds increment control. The FIFO high/low thresholds will be automatically increased by DPI_FIFO_INC when DPI internal FIFO is empty when ADP_FIFO_EN is enabled. Note the FIFO increment setting is in unit of entry.

TEST_SUM_SEL Select read value of RAL_RDATA/CKSUM. Refer to DPI_MON for more detail.

- **0** The read value is PAL_RDATA
- **1** The read value is CKSUM

+0048h DPI FIFO Read/Write Pointer Register DPI_ FIFO_PTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								FIFO_WPTR								
Type								RO								
Reset								0								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name								FIFO_RPTR								
Type								RO								
Reset								0								

Read/Write pointer of DPI internal FIFO.

© 2009 MediaTek Inc.

+004Ch DPI FIFO Valid Size Register DPI_ FIFO_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FIFO_SIZE									
Type							RO									
Reset							0									

The number of current DPI internal FIFO valid entries.

+0050h DPI FIFO Maximum Allowable High Threshold Register DPI_ FIFO_TH_MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FIFO_TH_MAX									
Type							R/W									
Reset							0									

This register is used for testing the maximum requirement of DPI internal FIFO size. Once FIFO_HIGH_TH is larger than FIFO_MAX in adaptive FIFO threshold control mode, interrupt with OVERSPEC status will be issued. Note the maximum settings of this register is 512.

+0054h DPI FIFO MAX Register DPI_FIFO_MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FIFO_MAX									
Type							R/W									
Reset							512									

FIFO_MAX Internal FIFO size, in unit of entry.

+0058h DPI Observe Register DPI_OBS_REG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												FIFO_AE_TH				
Type												RO				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							PRE_FIFO_SIZE									
Type							RO									

PRE_FIFO_SIZE Number of valid entries in internal FIFO after the current GMC transfer is complete.

FIFO_AE_TH The number of entries for the incoming GMC transfer.

© 2009 MediaTek Inc.

+0060h DPI Timing Generator Horizontal Control Register DPI_TGEN_HCNTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							DE_POL	HSYNC_POL	HFP							
Type							R/W	R/W	R/W							
Reset							0	0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HBP								HPW							
Type	R/W								R/W							
Reset	0								0							

DPI timing generator horizontal sync control register.

HPW Horizontal sync pulse width. (in pixel clock). For LCM with horizontal sync pulse width **6**, HPW should be set to **5** (6-1).

HBP Horizontal back porch width. (in pixel clock) For LCM with horizontal back porch width **6**, HBP should be set to **5** (6-1).

HFP Horizontal front porch width. (in pixel clock). For LCM with horizontal front porch width **6**, HFP should be set to **5** (6-1).

HSYNC_POL Hsync polarity

- **0** Negative
- **1** Positive

DE_POL Data enable polarity

- **0** Positive
- **1** Negative

+0064h DPI Timing Generator Vertical Control Register DPI_TGEN_VCNTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								VSYNC_POL	VFP							
Type								R/W	R/W							
Reset								0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VBP								VPW							
Type	R/W								R/W							
Reset	0								0							

DPI timing generator vertical sync control register

VPW Vertical sync pulse width. (in horizontal sync pulse). For LCM with vertical sync pulse width **6**, the VPW should be set to **5** (6-1).

VBP Vertical back porch width. (in horizontal sync pulse) For LCM with vertical back porch width **6**, the VBP should be set to **5** (6-1).

 © 2009 MediaTek Inc.

VFP Vertical front porch width. (in horizontal sync pulse) For LCM with vertical front porch width **6**, the VFP should be set to **5** (6-1).

VSYNC_POL Vsync polarity

0 Negative
1 Positive

+0070h DPI Hardware Cursor Control Register DPI_CUSR_CNTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CURSOR_LD															
Type	WO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CURSOR_SET		CURSOR_SIZE	CURSOR_EN
Type													R/W		R/W	R/W
Reset													0		0	0

The register controls the cursor format, set number and cursor pattern auto-load.

CURSOR_EN Hardware cursor enable. DPI_CNTL.FBC_EN and DPI_CNTL.DBS_EN shall be 0 if CURSOR_EN is on.

0 Disable
1 Enable

CURSOR_SIZE Select the cursor size

0 16x16
1 32x32

CURSOR_SET In 16x16 cursor size configuration, 4 set of cursors are supported. In 32x32 mode, this field is ignored.

0 Select cursor set 0
1 Select cursor set 1
2 Select cursor set 2
3 Select cursor set 3

CURSOR_LD Write 1 to this field will activate the cursor pattern DMA. The cursor pattern and palette will be copied from specified memory address to the internal cursor memory.

+0074h DPI Hardware Cursor Coordinate Register DPI_CUSR_CORD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							V_CORD									
Type							R/W									
Reset							0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							H_CORD									
Type							R/W									

Reset							0									

Set cursor set coordinate.

H_CORD Horizontal (x-axis) coordinate of cursor

V_CORD Vertical (y-axis) coordinate of cursor

+0078h DPI Hardware Cursor Memory Address Register DPI_CUSR_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DPI_CUSR_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DPI_CUSR_ADDR[15:0]															
Type	R/W															
Reset	0															

This register gives the memory address where the cursor data placed.

+0080h DPI Status Register DPI_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FB_INUSE		LINE_CNT									
Type					RO		RO									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												OUTEN	HWC_DMA_BUSY	TGEN_START	DP_START	BUSY
Type												RO	RO	RO	RO	RO

The read only register gives the current status of the DPI controller.

BUSY DPI is started

DP_START DPI data prepare is started

TGEN_START DPI timing generation is started

HWC_DMA_BUSY Hardware cursor DMA is ongoing

OUTEN Output pad of DPI signals is enabled to output

LINE_CNT Line number read of current frame

FB_INUSE The number of frame buffer being read

+0084h DPI Error Counter DPI_ERRCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DPI_ERRCNT									
Type							RO									

This register gives the pixel lost due to buffer underrun of RGB FIFO.

+0090h DPI Test Mode Register DPI_TMODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**DPI_OEN_ON**														
Type		R/W														
Reset		0														
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**PAL_RSEL**				**GMC_STATE**							**FIFO_TEST_OEN**	**FIFO_TEST_EN**	**MFIX_EN**	**IFIX_EN**	**OFIX_EN**
Type	R/W				RO							R/W	R/W	R/W	R/W	R/W
Reset	0											0	0	0	0	0

OFIX_EN Enable fix output instead of DPI input

- **0** Disable
- **1** Enable

IFIX_EN Enable fix input instead of GMC input

- **0** Disable
- **1** Enable

MFIX_EN Enable fix data instead of intermediate R/G/B FIFO output

- **0** Disable
- **1** Enable

FIFO_TEST_EN FIFO test enable. When this bit is enabled, the pop action of internal FIFO is taken control by FIFO_TEST_OEN.

- **0** Disable
- **1** Enable

FIFO_TEST_OEN FIFO test output enable. Each transition from 0 to 1 of this bit pops 1 entry out from internal FIFO.

GMC_STATE GMC state machine.

PAL_RSEL Palette select for read.

DPI_OEN_ON Turn on DPI output enable. It only takes effect when DPI_OEN_OFF and DSI_MODE are 0.

+0094h DPI Monitor Register DPI_MON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GMC_FULL**	**GMC_EMPTY**	**RPSH_ERR**	**GPSH_ERR**	**BPSH_ERR**	**RPOP_ERR**	**GPOP_ERR**	**BPOP_ERR**	**PAL_RDATA/CKSUM[23:16]**							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**PAL_RDATA/CKSUM[15:8]**								**PAL_RDATA/CKSUM[7:0]**							
Type	RO								RO							

PAL_RDATA / CKSUM The read data are selected by TEST_SUM_SEL.

PAL_RDATA Palette data read from BUS. Which palette data would be read can be selected by PAL_RSEL.

CKSUM Checksum value of DPI R/G/B bus of 1st frame whenever DPI_EN is enabled. The formula is as follows:

CKSUM[7:0](n) = CKSUM[7:0](n-1) + DPI_B[7:0](n)
CKSUM[15:8](n) = CKSUM[15:8](n-1) + DPI_G[7:0](n) + CKSUM[7](n-1)
CKSUM[23:16](n) = CKSUM[23:16](n-1) + DPI_R[7:0](n) + CKSUM[15](n-1)
CKSUM(0) = 24'b0, and the carry is ignored.

BPOP_ERR B FIFO pop error.
GPOP_ERR G FIFO pop error.
RPOP_ERR R FIFO pop error.
BPSH_ERR B FIFO push error.
GPSH_ERR G FIFO push error.
RPSH_ERR R FIFO push error.
GMC_EMPTY DPI internal FIFO empty.
GMC_FULL DPI internal FIFO full.

5.8.2.2 DPI Clock Settings Examples

The frequency of the pixel clock in the DPI controller is depending on the panel display resolution and the pixel clock generation PLL.

5.8.2.3 DPI Register Settings Examples

Here is an example for setting the DPI controller registers. In the following figure, consider a virtual frame buffer **FB0** in 24bpp format with resolution 1024 * 768 and an LCD panel with resolution 320 * 240. The virtual frame buffer resides at the external memory address 0x3010_0000 and we would like to display the rectangle area (x_0, y_0) to (x_0+h_size, y_0+v_size) in the virtual frame buffer. The following register values should be calculated and set by software:

1. DPI_SIZE: h_size = 319, v_size = 239
2. DPI_FB0_ADDR: 0x3100_0000 + y_0 * 1024 * 3 + x_0 * 3
3. FB0_LINE_STEP: 1024 * 3 (24bpp) = 3072
4. DPI_TGEN_HCNTL: refer to the specification of your LCD panel module and fill the correct register values.
5. DPI_TGEN_VCNTL: the same as DPI_TGEN_VCNTL
6. Enable the DPI module by setting both the register bits DPI_CNTL[1] and DPI_CNTL[0] to 1.

 © 2009 MediaTek Inc.

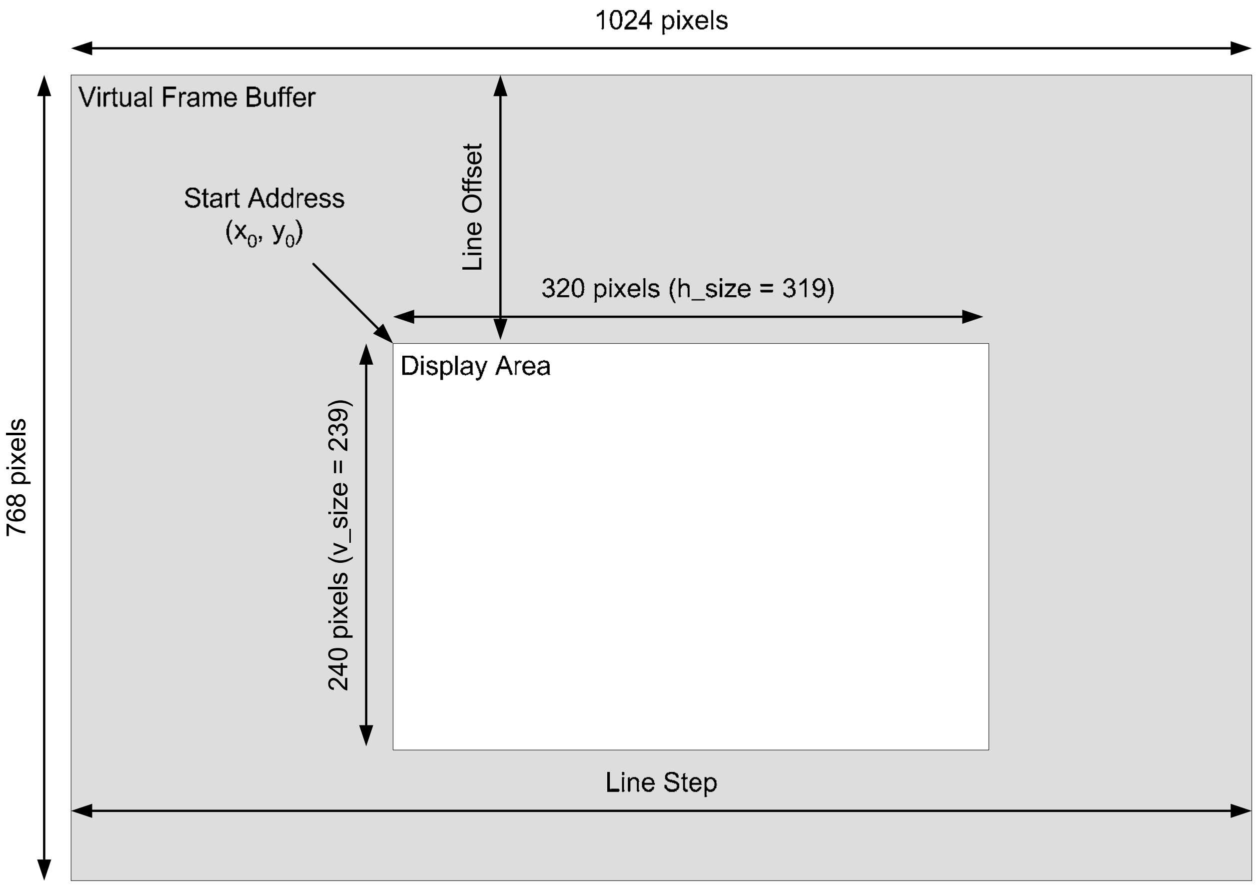


Figure 119 ***A frame buffer example for DPI register settings.***

5.8.2.4 Hardware Cursor Settings

To enable the hardware cursor, user should prepare the appropriate cursor data in the memory and trigger the DPI controller to move the cursor data in. **Figure 120** shows the cursor memory organization. The pixel data is arranged in the scan line order. For a 32x32 hardware cursor, the pixel data occupies 32*32*0.5 = 512 bytes. For a 16x16 cursor, the pixel data occupies 16*16*0.5 = 128 bytes. Thus the four 16x16 cursor pixel data should be put consecutively in the 512 byte memory space.

Since the pixel data stored in the 4-bit indexed color mode, a color palette is needed. For memory alignment, the 24-bit RGB value is stored in an 32-bit word with 8-bit MSB not used. Moreover, the palette contains only 14 colors since the color index 0 is reserved as the transparent color key and color index 1 is used as the inverse of background color.

Suppose we would like to use the 16x16 cursor set 1, the hardware cursor could be enabled in the following steps:

1. DPI_CUSR_ADDR: set to the correct address of the cursor memory.
2. DPI_CUSR_CNTL: set bit 31 to 1 for auto moving the cursor memory into the DPI controller.

3. DPI_CUSR_CORD: set the coordinate of the left-upper corner of the cursor
4. DPI_CUSR_CNTL: set CURSOR_SET to 1, CURSOR_FMT to 0, and CURSOR_EN to 1

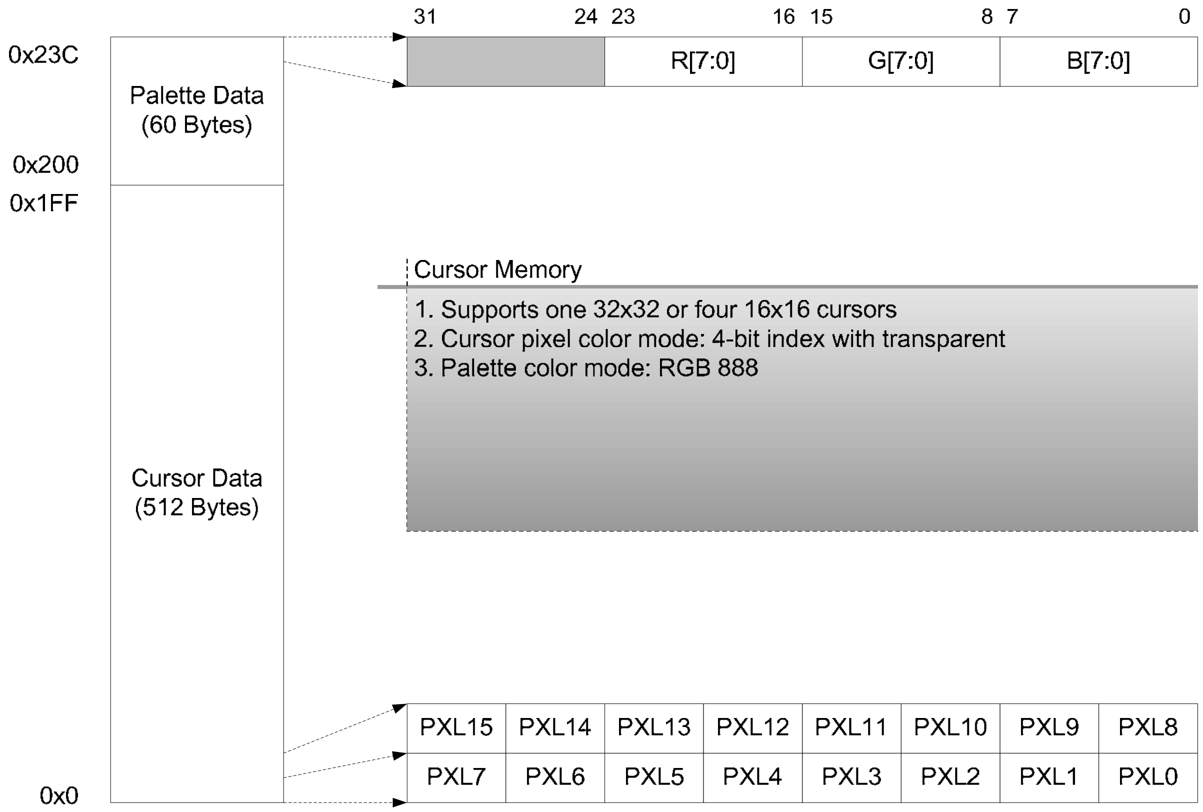


Figure 120 ***Cursor memory organization.***

5.9 Drop Resize

5.9.1 General Description

This block provides a simple scaling down function by performing pixel and line dropping. It receives data, performs the image resizing function and outputs to the image process engine module. It can scale down the input image by any ratio > 1/2047. However, the maximum sizes of input and output images are limited to 4096x4096.

5.9.2 Register Definitions

5.9.2.1 Register Map

Table 100 shows the register map.

REGISTER ADDRESS	REGISTER NAME	SYNONYM
DRZ+ 0000h	Drop Resize Start Register	**DRZ_STR**
DRZ+ 0004h	Drop Resize Control Register	**DRZ_CON**

 © 2009 MediaTek Inc.

DRZ + 0008h	Drop Resize Status Register	**DRZ_STA**
DRZ + 000Ch	Drop Resize Interrupt Acknowledge Register	**DRZ_ACKINT**
DRZ + 0010h	Drop Resize Source Image Size Register	**DRZ_SRC_SIZE**
DRZ + 0014h	Drop Resize Target Image Size Register	**DRZ_TAR_SIZE**
DRZ + 0020h	Drop Resize Horizontal Ratio Register	**DRZ_RAT_H**
DRZ + 0024h	Drop Resize Vertical Ratio Register	**DRZ_RAT_V**

Table 100 ***Register map.***

5.9.2.2 Register Description

Followings are detail descriptions of each register.

DRZ+0000h Drop Resize Start Register DRZ_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of Drop Resize. Note that before setting STR to "1", all the configurations shall be done by giving proper values.

STR Start the Drop resize engine. Write 1 to this bit will start the FSM of Drop resize. Write 0 to this bit will reset the FSM of Drop resize.

DRZ+0004h Drop Resize Configuration Register DRZ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**AUTO RSTR**			**IT**
Type													R/W			R/W
Reset													0			0

The register specifies the configuration of Drop resize.

IT Interrupt Enabling

- **0** Disable
- **1** Enable

AUTO RSTR Automatic restart. Drop Resize automatically restarts itself while current frame is finished.

- **0** Disable
- **1** Enable

DRZ+0008h Drop Resize Status Register DRZ_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**RUN**
Type																RO
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**IT**
Type																RO
Reset																0

This register helps software program being well aware of the global status of Drop Resize.

IT Interrupt status for Drop Resize
- 0 No interrupt is generated.
- 1 An interrupt is pending and waiting for service.

RUN Drop Resize status
- 0 Drop Resize is stopped or has completed the transfer already.
- 1 Drop Resize is currently running.

DRZ+000Ch Drop Resize Interrupt Acknowledge Register DRZ_ACKINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ACK**
Type																WO

This register is used to acknowledge the current interrupt request associated with the completion event of Drop Resize by software program. Note that this is a write-only register, and any read to it will return a value of “0”.

ACK Interrupt acknowledge for the Drop Resize
- 0 No effect
- 1 Interrupt request is acknowledged and should be relinquished.

DRZ+0010h Drop Resize Source Image Size Register DRZ_SRC_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**V[11:0]**											
Type					R/W											
Reset					X											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**H[11:0]**											
Type					R/W											
Reset					X											

The register specifies the size of source image. **The maximum allowable size is 4096x4096**.

V the height of source image-1

H the width of source image-1

DRZ+0014h Drop Resize Target Image Size Register DRZ_TAR_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**V[11:0]**											
Type					R/W											
Reset					0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**H[11:0]**											
Type					R/W											
Reset					0											

The register specifies the size of target image. **The maximum allowable size is 4096x4096**.

V the height of target image-1

H the width of target image-1

DRZ+0020h Drop Resize Horizontal Ratio Register DRZ_RAT_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						**I [10:0]**										
Type						R/W										
Reset						X										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**Q [11:0]**											
Type					R/W											
Reset					X											

The register specifies horizontal resizing ratio. It is obtained by (the width of source image/the width of target image) = I + Q/P = I + Q/the width of target image.

I the integer part

Q the denominator

DRZ+0024h Drop Resize Vertical Ratio Register DRZ_RAT_V

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						**I [10:0]**										
Type						R/W										
Reset						X										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**Q [11:0]**											
Type					R/W											
Reset					X											

The register specifies horizontal resizing ratio. It is obtained by (the height of source image/the height of target image) = I + Q/P = I + Q/the height of target image.

I the integer part

Q the denominator

5.10 Display Serial Interface Controller

5.10.1 Features

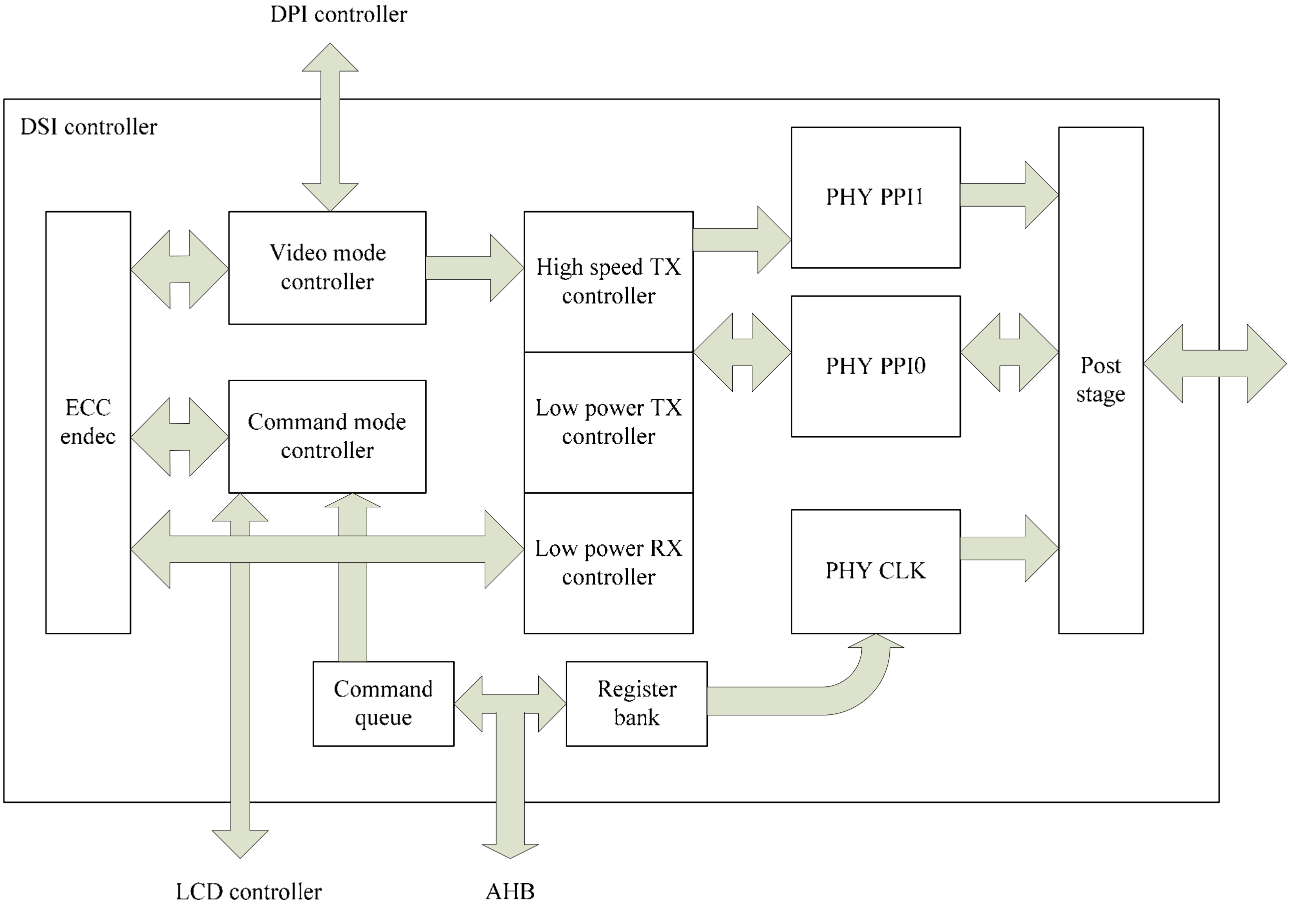
MT6516 DSI controller is compliant to DSI specification v1.01r5 and D-PHY specification v0.9. The display serial interface controller supports two lanes with up-to 540Mb/s for each. Both command mode and video

mode are provided for flexible link to versatile MIPI compliant display. The DSI controller also supports both high speed and low power transmission for forward link and low power transmission for reverse link. In command mode, a dedicated command queue is designed for easy SW manipulation of all possible DSI commands and transmission modes for batch transfer. In video mode, all traffic sequences and all packet stream formats are supported.

- Compliant to DSI specification v1.01r5
- Two lanes, 540Mb/sec for each
- Command, and video modes
- High speed, low power forward transmission
- Bus-Turn-Around and Low power reverse transmission
- ECC and Checksum Capabilities
- Video packet stream: 16-bpp, 18-bpp, packed 18-bpp, and 24-bpp
- Video traffic sequences: sync pulse, sync event and burst modes
- Multiple Peripheral Support
- Command queue

 © 2009 MediaTek Inc.

5.10.2 Block Diagram



5.10.3 Control Registers

5.10.3.1 Register Definition of DSI Controller

DSI_BASE in MT6516 is 0x80140000.

+0000h DSI Start Register DSI_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DSI_START
Type																R/W
Reset																0

DSI_START Start DSI controller operation

0 DSI controller off

1 DSI controller on

+0004h DSI Status DSI_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																BUSY
Type																R
Reset																0

BUSY DSI busy status

0 DSI controller is idle.

1 DSI controller is busy.

+0008h DSI Interrupt Enable Register DSI_INTEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DSI_OUT_DIS						CMD_DONE	RD_RDY
Type									R/W						R/W	R/W
Reset									0						0	0

RD_RDY DSI command mode read data ready interrupt enable. It is recommended to turn on this interrupt if there are read commands in command queue. After read command is send by DSI TX, the read response will be received in a short time. To avoid the read data being overwritten by next read response when there are multiple read commands in command queue, the interrupt will notify SW to

read the data and make sure it's ready to receive next read response packet by setting MCU_RACK before leaving the interrupt service routine.

0 Disable

1 Enable

CMD_DONE DSI command queue finish interrupt enable. After all commands in command queue have been done, the interrupt will be issued.

0 Disable

1 Enable

DSI_OUT_DIS DSI output disable.

0 DSI output enable. When DSI active, this bit should be clear to 0.

1 DSI output disable. When DSI disable, this bit should be set to 1 to power down MIPI analog macro.

+000Ch DSI Interrupt Enable Register DSI_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															CMD_DONE	RD_RDY
Type															R	R
Reset															0	0

This read only register gives the current interrupt request status of the DSI controller. Each field is corresponded to the previous DSI_INTEN register.

+0010h DSI Common Control Register DSI_COM_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DSI_RESET
Type																R/W
Reset																0

DSI_RESET DSI software reset

0 De-assert DSI software reset.

1 Assert DSI software reset.

+0014h DSI Mode Control Register DSI_MODE_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**MODE_CON**	
Type															R/W	
Reset															0	

MODE_CON DSI mode control

- **00** Command mode.
- **01** Sync-pulse video mode.
- **10** Sync-event video mode.
- **11** Burst video mode.

+0018h DSI TX RX Control Register DSI_TXRX_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**MAX_RTN_SIZE**					**CORR_EN**	**CKSM_EN**	**ECC_EN**	**NULL_EN**			**LANE_NUM**			**VC_NUM**	
Type	R/W					R/W	R/W	R/W	R/W			R/W			R/W	
Reset	0					0	0	0	0			1			0	

VC_NUM Virtual channel number in video mode

LANE_NUM Lane number

- **00** Not allowed.
- **01** One lane.
- **10** Two lanes.
- **others** Not allowed.

NULL_EN Enable null packet transfer in BLLP

- **0** Disable.
- **1** Enable.

ECC_EN Enable receive packet ECC decode

- **0** Disable.
- **1** Enable.

CKSM_EN Enable receive packet checksum calculation

- **0** Disable.
- **1** Enable.

CORR_EN Enable receive packet error correction

- **0** Disable.
- **1** Enable.

MAX_RTN_SIZE Maximum return packet size. This register constrains maximum return packet that slave side will send back to host. It will take effect after host send "Set Maximum Return Packet Size" packet to slave.

+001Ch DSI Pixel Stream Control Register DSI_PSCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DSI_PS_SEL**		**DSI_PS_WC**											
Type			R/W		R/W											
Reset			0		0											

PS_WC The word count of the long packet in valid pixel data duration in unit of byte. This value must be (H_SIZE*BPP). Take the QVGA display as an example, the value of PS_WC is (240*3) = 720 in decimal.

PS_SEL Select the pixel stream type, please also refer to DSI specification.

- **00** Packed pixel stream with 16-bit RGB 5-6-5 format.
- **01** Loosely Pixel stream with 18-bit RGB 6-6-6 format.
- **10** Packed pixel stream with 24-bit RGB 8-8-8 format.
- **11** Packed pixel stream with 18-bit RBG 6-6-6 format.

+0020h DSI Vertical Sync Active Register DSI_VSA_NL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**VSA_NL**						
Type										R/W						
Reset										0						

VSA_NL Vertical sync active duration which is in unit of line.

+0024h DSI Vertical Back Porch Register DSI_VBP_NL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**VBP_NL**						
Type										R/W						
Reset										0						

VBP_NL Vertical back porch duration which is in unit of line.

+0028h DSI Vertical Front Porch Register DSI_VFP_NL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**VFP_NL**						
Type										R/W						
Reset										0						

VFP_NL Vertical front porch duration which is in unit of line.

+002Ch DSI Vertical Active Register DSI_VACT_NL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							**VACT_NL**									
Type							R/W									
Reset							0	0	0	0	0	0	0	0	0	0

VACT_NL Vertical active duration which is in unit of line.

+0030h DSI Line Byte Register DSI_LINE_NB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					**DSI_LINE_NB**											
Type					R/W											
Reset					0											

LINE_NB Line duration in unit of byte. This value is (HPW + HBP + RGB + HFP) * BPP in Sync Pulse Mode and (HBP + RGB_HFP) * BPP in Sync Event Mode with or without Burst Mode. Please also refer to section 5.10.4.

+0034h DSI Horizontal Sync Active Byte Register DSI_HSA_NB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					**DSI_HSA_NB**											
Type					R/W											
Reset					0											

HSA_NB Horizontal sync active duration in unit of byte. This register is needed to be set in Sync Pulse Mode only. The value is (HPW * BPP – 4). Please also refer to section 5.10.4.

+0038h DSI Horizontal Back Porch Byte Register DSI_HBP_NB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					**DSI_HBP_NB**											
Type					R/W											
Reset					0											

HBP_NB Horizontal back porch duration in unit of byte. The value is (HBP * BPP) in Sync Pulse Mode and is (HBP * BPP – 4) in Sync Event Mode and Burst Mode. Please also refer to section 5.10.4.

+003Ch DSI Horizontal Front Porch Byte Register DSI_HFP_NB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DSI_HFP_NB											
Type					R/W											
Reset					0											

HFP_NB Horizontal front porch duration in unit of byte. The value is (HFP * BPP – 6). Please also refer to section 5.10.4.

+0040h DSI RGB Byte Register DSI_RGB_NB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DSI_RGB_NB											
Type					R/W											
Reset					0											

RGB_NB Line valid pixel data duration in unit of byte. It is the length of pixel stream data including packet header, packet data (payload), and packet footer. The value is (RGB * BPP + 6). Please also refer to section 5.10.4.

+0050h DSI Horizontal Sync Active Word Count Register DSI_HSA_WC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DSI_HSA_WC											
Type					R/W											
Reset					0											

HSA_WC The word count of the long packet in horizontal sync active duration. This register must be (HSA_NB – 6). Please also refer to section 1.4.

+0054h DSI Horizontal Back Porch Word Count Register DSI_HBP_WC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DSI_HBP_WC											
Type					R/W											
Reset					0											

© 2009 MediaTek Inc.

HBP_WC The word count of the long packet in horizontal back porch duration. This register must be (HBP_NB – 6). Please also refer to section 1.4.

+0058h DSI Horizontal Front Porch Word Count Register DSI_HFP_WC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					DSI_HFP_WC											
Type					R/W											
Reset					0											

HFP_WC The word count of the long packet in horizontal front porch duration. This register must be (HFP_NB – 6). Please also refer to section 1.4.

+0060h DSI Command Queue Control Register DSI_CMDQ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name									CMDQ_SEL			CMDQ_SIZE				
Type									R/W			R/W				
Reset									0			0				

CMDQ_SIZE Number of commands in command queue.
CMDQ_SEL Select the current usage of command queue. There are two dedicated command queue for DSI controller.

Ps. The usage of the DSI command queue is described in section 5.10.7 in detail.

+0070h DSI Receiver Status Register DSI_RX_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																DIR
Type																R
Reset																0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		TRIG	DEC2_EN	MB_ERR2	SB_ERR2	DEC_OK2	CKSM_ERR	MB_ERR1	SB_ERR1	DEC_OK1	PKT_LEN				LONG	BUSY
Type		R	R	R	R	R	R	R	R	R	R				R	R
Reset		0	0	0	0	0	0	0	0	0	0				0	0

BUSY DSI controller is receiving data.
LONG Received a long packet.
PKT_LEN Received packet length in unit of byte.
DEC_OK1 Received packet is decoded and no error found.
SB_ERR1 Received packet is decoded and has single-bit error.
MB_ERR1 Received packet is decoded and has multiple-bit error.

CKSM_ERR Received long packet has check-sum error.
DEC_OK2 Received acknowledge with error report packet is decoded and no error found.
SB_ERR2 Received acknowledge with error report packet is decoded and has single-bit error.
MB_ERR2 Received acknowledge with error report packet is decoded and has multiple-bit error.
DEC2_EN Received acknowledge with error report packet is decoded.
TRIG Received low level trigger message. If this bit is 1, please ignore the other status and read DSI_TRIG_STA for determine the trigger command.
DIR Current bus direction of Data Lane 0. If this bit is 1, there is reverse direction transmissions on Data Lane 0 in Low Power Mode. Otherwise, it is a forward direction transmissions.

+0074h DSI Receive Packet Data Byte 0 ~ 3 Register DSI_RX_DATA03

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BYTE3								BYTE2							
Type	R								R							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BYTE1								BYTE0							
Type	R								R							
Reset	0								0							

Data bytes 0 ~ 3 received from slave.

+0078h DSI Receive Packet Data Byte 4 ~ 7 Register DSI_RX_DATA47

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BYTE7								BYTE6							
Type	R								R							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BYTE5								BYTE4							
Type	R								R							
Reset	0								0							

Data bytes 4 ~ 7 received from slave.

+007Ch DSI Receive Packet Data Byte 8 ~ B Register DSI_RX_DATA8B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BYTEB								BYTEA							
Type	R								R							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BYTE9								BYTE8							
Type	R								R							
Reset	0								0							

Data bytes 8 ~ B received from slave.

© 2009 MediaTek Inc.

+0080h DSI Receive Packet Data Byte C Register DSI_RX_DATAC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BYTEC**							
Type									R							
Reset									0							

Data byte C received from slave.

+0084h DSI Read Data Acknowledge Register DSI_RACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RACK**
Type																WC
Reset																0

When current read command is executed and requested read data is completely received by DSI controller, the IRQ with RD_RDY status will be issued to inform SW to get the receiving status and received data from register DSI_RX_STA and DSI_RX_DATA* respectively. It's mandatory to write this register to keep doing next command.

+0088h DSI Trigger Message Status Register DSI_TRIG_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**TRIG_3**	**TRIG_2**	**TRIG_1**	**TRIG_0**
Type													R	R	R	R
Reset													0	0	0	0

TRIG_0 Reserved by DSI specification.
TRIG_1 Acknowledge.
TRIG_2 TE.
TRIG_3 Remote Application Reset.

+0090h DSI Memory Continue Command Register DSI_MEM_CONTI

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DSI_WMEM_CONTI**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	DSI_RMEM_CONTI
Type	R/W
Reset	0

RMEM_CONTI Read memory continue command.

WMEM_CONTI Write memory continue command.

+0094h DSI Frame Byte Count Register DSI_FRM_BC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												FRM_BC[20:16]				
Type												R/W				
Reset												0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FRM_BC[15:0]															
Type	R/W															
Reset	0															

FRM_BC The total number of byte is expected to be read for type-3 command. Please also refer to section 5.10.7 for the usage of type-3 command.

+0100h DSI PHY Control Register DSI_PHY_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BTA_TO_RST	LRX_TO_RST	HTX_TO_RST					PHY_RST
Type									R/W	R/W	R/W					R/W
Reset									0	0	0					0

PHY_RST Reset PHY related control circuit.

HTX_TO_RST High speed TX time-out reset. Force current HSTX finish and go to LP-11 state.

LRX_TO_RST Low power RX time-out reset. Abort current LPRX finish and go to LP-11 state.

BTA_TO_RST Bus-turn-around time-out reset. Abort current BTA and go to LP-11 state

+0104h DSI PHY Lane Clock Control Register DSI_PHY_LCCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														LC_WAKEUP_EN	LC_ULPM_EN	LC_HS_TX_EN
Type														R/W	R/W	R/W
Reset														0	0	0

LC_HS_TX_EN Start clock lane high speed transmission.

LC_ULPM_EN Make the clock lane go to ultra-low power mode.

LC_WAKEUP_EN Make the clock lane wake-up from ultra-low power mode.

Ps. Please also refer to section 5.10.4 for the register setting sequence to enter/exit ultra-low power mode

+0108h DSI PHY Lane 0 Control Register DSI_PHY_LD0CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														L0_WAKEUP_EN	L0_ULPM_EN	L0_RM_TRIG_EN
Type														R/W	R/W	R/W
Reset														0	0	0

L0_RM_TRIG_EN Send application trigger to slave side.

L0_ULPM_EN Make the data lane 0 go to ultra-low power mode.

L0_WAKEUP_EN Make the data lane 0 wake-up from ultra-low power mode.

+0110h DSI PHY Timing Control 0 Register DSI_PHY_TIMCON0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HS_TRAIL								HS_ZERO							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HS_PRPR								LPX							
Type	R/W								R/W							
Reset	0								0							

Please refer to section 5.10.5 for more detail. All these timing are counted by dsi_nx_ck, and the frequency of dsi_nx_ck is specified by DSI_PHY_ANACON0.RG_NX_CK_SEL, DSI_PHY_ANACON0.PLL_DIV1, DSI_PHY_ANACON1.PLL_DIV2, and DSI_PHY_ANACON1.PLL_CLKR, please refer to figure 1.

+0114h DSI PHY Timing Control 1 Register DSI_PHY_TIMCON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TA_SACK								TA_GET							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TA_SURE								TA_GO							
Type	R/W								R/W							
Reset	0								0							

Please refer to section 5.10.5 for more detail. All these timing are counted by dsi_nx_ck, and the frequency of dsi_nx_ck is specified by DSI_PHY_ANACON0.RG_NX_CK_SEL, DSI_PHY_ANACON0.PLL_DIV1, DSI_PHY_ANACON1.PLL_DIV2, and DSI_PHY_ANACON1.PLL_CLKR, please refer to figure 1.

+0118h DSI PHY Timing Control 2 Register DSI_PHY_TIMCON2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CLK_TRAIL								CLK_ZERO							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LPX_WAIT								CONT_DET							
Type	R/W								R/W							
Reset	0								0							

Please refer to section 5.10.5 for more detail. All these timing are counted by dsi_nx_ck, and the frequency of dsi_nx_ck is specified by DSI_PHY_ANACON0.RG_NX_CK_SEL, DSI_PHY_ANACON0.PLL_DIV1, DSI_PHY_ANACON1.PLL_DIV2, and DSI_PHY_ANACON1.PLL_CLKR, please refer to figure 1.

+011Ch DSI PHY Timing Control 3 Register DSI_PHY_TIMCON3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CLK_HS_PRPR							
Type									R/W							
Reset									0							

Please refer to section 5.10.5 for more detail. All these timing are counted by dsi_nx_ck, and the frequency of dsi_nx_ck is specified by DSI_PHY_ANACON0.RG_NX_CK_SEL, DSI_PHY_ANACON0.PLL_DIV1, DSI_PHY_ANACON1.PLL_DIV2, and DSI_PHY_ANACON1.PLL_CLKR, please refer to figure 1.

 © 2009 MediaTek Inc.

5.10.3.2 Register Definition of Analog PHY

PLL_BASE in MT6516 is 0x80060000.

80060b00h DSI Analog PHY Control 0 Register DSI_PHY_ANA CON0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PHY_BIST_MODE	DSI_BIST_EN	DSI_FIX_PAT	RG_PLL_DIV1						RG_NX_CK_SEL		RG_CK_SEL		RG_PLL_LN		PLL_EN
Type	R/W	R/W	R/W	R/W						R/W		R/W		R/W		R/W
Reset	0	0	0	001110						00		00		00		0

PLLEN PLL CLK enable

PLL_LN Lane number, please see figure 1

CK_SEL CK selection for DSI controller, please see figure 1

NX_CK_SEL CK selection for DSI PHY, please see figure 1

PLL_DIV1 VCO output frequency (Fvco) = reference_clock*DIV1, please see figure 1

DSI_FIX_PAT Make DSI to send fixed test pattern when BIST test. The fixed test pattern sequence is 0x00, 0x01, 0x02, 0x04, 0x08, 0xff, 0xfe, 0xfd,0xfb, 0xf7, 0x00, 0xff, 0x55, 0xaa, 0x88, 0xff.

DSI_BIST_EN BIST test enable

PHY_BIST_MODE BIST test enable. DSI_BIST_EN and PHY_BIST_MODE must be set to 1 when BIST mode

80060b04h DSI Analog PHY Control 1 Register DSI_PHY_ANA CON1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RG_LNT_LOOPBACK			RG_PLL_CLKR	RG_PLL_CLF		RG_PLL_CVCOB		RG_PLL_CCP				RG_PLL_DIV2			
Type	R/W			R/W	R/W		R/W		R/W				R/W			
Reset	000			0	01		00		0101				0000			

PLL_DIV2 Please see figure 1.1

PLL_CCP PLL CP control

PLL_CVCOB PLL VCO bias control

PLL_CLF PLL VCO loop filter

PLL_CLKR Selecting reference clock, please see figure 1

- **0** reference clock = 26MHz
- **1** reference clock = 13MHz

LNT_LOOPBACK Lane Loopback mux control

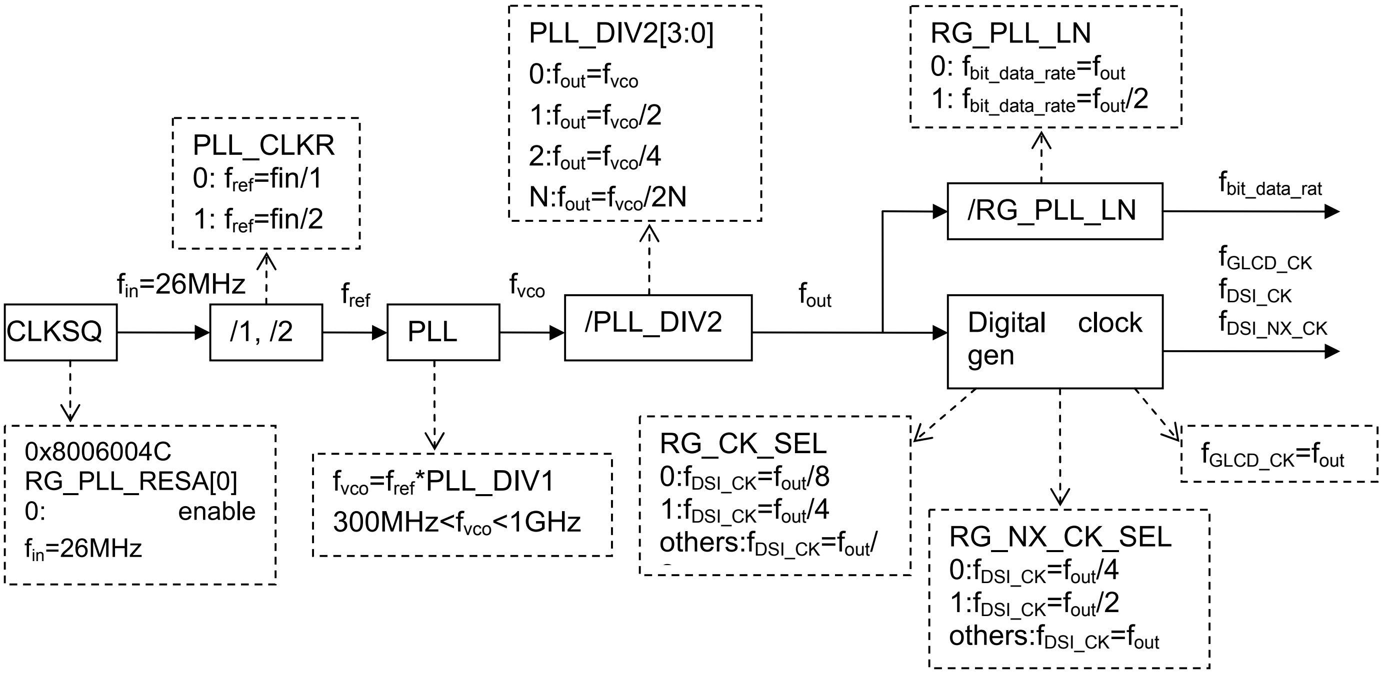


*Figure 121 **clock structure of MIPI macro***

80060b08h DSI Analog PHY Control 2 Register — DSI_PHY_ANACON2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		RG_LNT_LPCD_CALI			RG_LNT_HSRX_CALI			RG_LNT_BGR_DOUT2_SEL		RG_LNT_BGR_DOUT1_SEL		RG_LNT_BGR_DIV		RG_LNT_BGR_SELPH	RG_LNT_BGR_CHPEN	RG_LNT_BGR_EN
Type		R/W			R/W			R/W		R/W		R/W		R/W	R/W	R/W
Reset		010			010			0		0		0		0	1	1

RG_LNT_BGR_EN BGR enable

RG_LNT_BGR_CHPEN BGR chop enable

RG_LNT_BGR_SELPH BGR clock phase sel

RG_LNT_BGR_DIV BGR clock div

RG_LNT_BGR_DOUT1_SEL BGR debug output1 sel

RG_LNT_BGR_DOUT2_SEL BGR debug output2 sel
RG_LNT_HSRX_CALI Lane HSRX calibration
RG_LNT_LPCD_CALI Lane LPCD calibration

80060b0ch DSI Analog PHY Control 3 Register DSI_PHY_ANACON3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RG_LNT_LPTX_CALI				RG_LP_BIAS_EN	RG_LNT_AIO_SEL				RG_LNT_CIRE		RG_LNT_CALZ_CZ				RG_LNT_CALZ_EN
Type	R/W				R/W	R/W				R/W		R/W				R/W
Reset	010				0	000				01		0110				0

RG_LNT_CALZ_EN Lane enable HS impedance calibration. Active high
RG_LNT_CALZ_CZ Lane HS impedance calibration
RG_LNT_CIRE Lane VREG (output swing) control. 00: 0.38V; 01: 0.4V; 10: 0.42V; 11: 0.44V
RG_LNT_AIO_SEL Lane Analog I/O debug sel
RG_LP_BIAS_EN Lane LP bias enable
RG_LNT_LPTX_CALI Lane LPTX slew rate calibration

80060b10h DSI Analog PHY Control 4 Register DSI_PHY_ANACON4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			RG_LNT1_HSTX_SPT	RG_LNT0_HSTX_SPT	RG_LNT1_HS_CZ				RG_LNT0_HS_CZ				RG_LNTC_HS_CZ			
Type			R/W	R/W	R/W				R/W				R/W			
Reset			0	0	0110				0110				0110			

RG_LNTC_HS_CZ Lane CLK HS impedance calibration
RG_LNT0_HS_CZ Lane0 HS impedance calibration
RG_LNT1_HS_CZ Lane1 HS impedance calibration
RG_LNT0_HSTX_SPT Lane0 HSTX sync sample point
RG_LNT1_HSTX_SPT Lane1 HSTX sync sample point

80060b14h DSI Analog PHY Control 5 Register DSI_PHY_ANACON5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESERVED**					**FGLCD_SEL**	**MIPI_OUT_ISO_EN**	**PAT_EN**	**MIPI_ISO**	**RG_LNT_HSTX_EDGE_SEL**	**RG_TMODE**	**RG_FORCE_TX_D**			**RG_FORCE_EN**	
Type	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	
Reset	0	0	0	0		0	0	0	0	0	0	000			00	

FORCE_EN Force enable:'00'=original '01'=HSTX mode '10'=LPTX mode '11'=RX Mode

- **00** original
- **01** HSTX mode
- **10** LPTX mode
- **11** RX mode

FORCE_TX_D TX DATA

- **000** original
- **001** LP all 1
- **010** LP all 0
- **011** reserved
- **100** HS PRBS
- **101** HS all 1
- **110** HS all 0
- **111** reserved

TMODE Analog Test Mode, '0'=disable, '1'=enable

- **0** disable
- **1** enable

LNT_HSTX_EDGE_SEL Select the synchronization between clock lane & 1st bit of one word.

- **0** sync 1st bit with rising edge
- **1** sync with falling edge

MIPI_ISO MIPI isolation test mode. All MIPI PHY input output ports are connected to chip pads

PAT_EN Enable DSI to send test patterns specified in MIPITX_CON6 and MIPITX_CON7.

MIPI_OUT_ISO_EN Isolate MIPI macro output when MIPI macro power down.

FGLCD_SEL Clock selection

- **0** fglcd_ck, fdsi_ck and fdsi_nx_ck come from MIPI macro output.
- **1** fglcd_ck, fdsi_ck and fdsi_nx_ck all come from clock square 26MHz.

RESERVED Reserved

80060b18h MIPITX Configuration 6 MIPITX_CON6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIST_PATTERN[15:0]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

80060b1ch MIPITX Configuration 7 MIPITX_CON7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BIST_PATTERN[31:16]**															

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIST_PATTERN DSI will send this BIST pattern[31:0] again and again when MIPITX_CON5.PAT_EN = 1.

5.10.4 Clock Control

In MT6516, the clock source of DSI and D-PHY are driven from internal PLL. Before turning on this PLL, all DSI and D-PHY circuits are inactive. To enable DSI display, it's necessary to enable the PLL first and adjust the PLL to output a suitable clock frequency. Since our internal PLL maximum clock frequency is 540 Mhz which cannot support to operate on one data lane for 24-bpp VGA resolution in 60fps. Table. 5-1 gives an example to show how to select a suitable clock frequency.

Table. 5-1: Clock related parameters of VGA display

Pixel rate (Mhz)	**BPP**	**Bit clock frequency (Mhz)**	**Required PLL output clock frequency (Mhz)**		**rg_pll_ln** (PHY_ANDCON1[2:1])	**Lane Num** (TXRX_CON [4:2])	**D-PHY cycle time[*1] (ns) for rg_nx_ck_sel** (PHY_ANACON1[6:5])		
							0	1	2
25	2	25*2*8 = 400	1 lane	400	0	1	2.5	5	7.5
			2 lanes	400	1	2	2.5	5	7.5
				200	0	2	5	10	15
	3	25*3*8 = 600	1 lane	600 (N/A)	-	-	-	-	-
			2 lanes	300	0	2	3.3	6.7	10

[*1]D-PHY cycle time is selected by rg_nx_ck_sel. D-PHY clock is equal to PLL output clock divided by $2^{rg_nx_ck_sel}$. The registers DSI_TIMCON0 ~ DSI_TIMCON3 are counted by this clock.

After enabling the clock and properly setting the related registers, it's necessary to the enable clock lane before starting high speed data transmission by setting the DSI_PHY_LCCON[0] as 1. If we are going to stop high speed transmission enter ultra-low power mode, the register sequence is described as Table. 5-2. The wake-up sequence is also shown in Table. 5-3. For more detail about the relationship between register setting and D+/D- on the lane, please see the timing diagram illustrated in Fig. 5-1 and Fig. 5-2.
Similar to clock lane, data lane 0 also supports ultra-low power mode. For some reason, data lane 0 may need to enter ultra-low power mode, it should be prior to clock lane. In MT6516, entering ultra-low power mode on data lane has no essential power saving on host side.

Table. 5-2: The sequence to enter ultra-low power mode on clock lane

Step	Description	Register Setting
I	Stop the high speed transmission on clock lane	Write DSI_PHY_LCCON[0] = 0
II	Enable ultra-low power mode	Write DSI_PHY_LCCON[1] = 1
III	Disable PLL clock	Write PHY_ANACON1[0] (rg_pll_en) = 0

Table. 5-3: The sequence to exit ultra-low power mode on clock lane

Step	Description	Register Setting
I	Turn-on PLL clock	Write PHY_ANACON1[0] (rg_pll_en) = 1

Step	Description	Register Setting
II	Wait for PLL stable, than disable ultra-low power mode	Write DSI_PHY_LCCON[1] = 0
III	Wait for at least 1ms, enable wake-up	Write DSI_PHY_LCCON[2] = 1
IV	Disable wake-up	Write DSI_PHY_LCCON[2] = 0
V	Enable high speed clock if necessary	Write DSI_PHY_LCCON[0] = 1

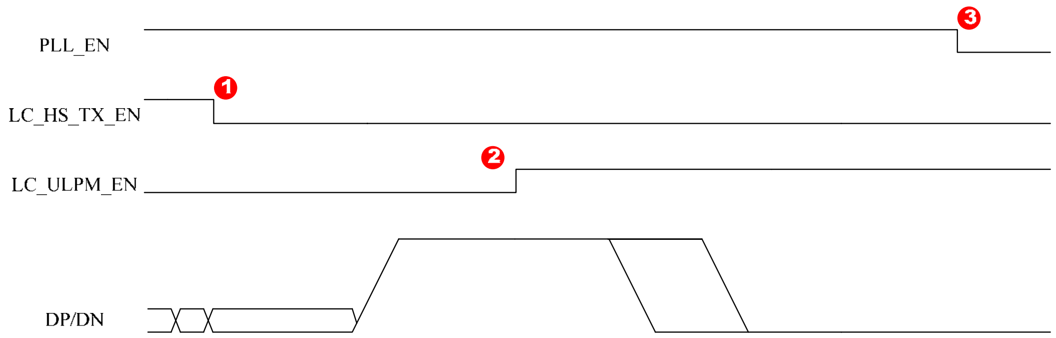


Fig. 5-1: Sequence of entering ultra-low power mode on clock lane

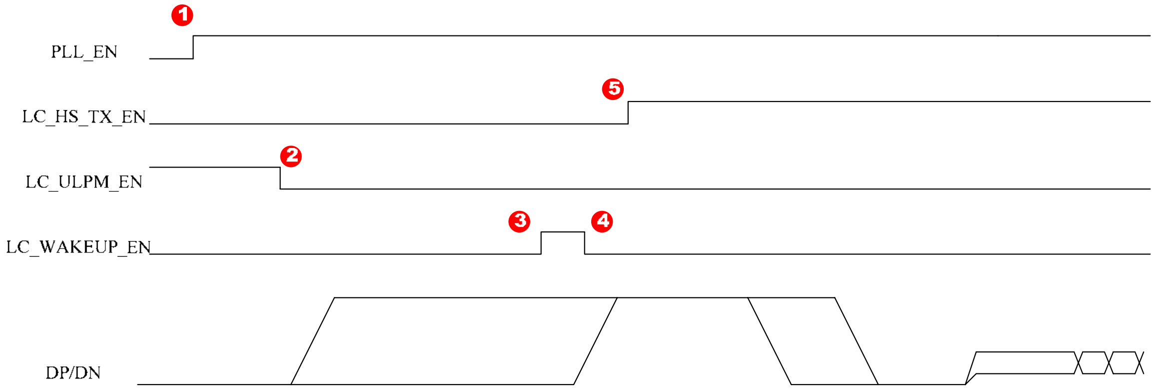


Fig. 5-2: Sequence of exiting ultra-low power mode on clock lane

© 2009 MediaTek Inc.

5.10.5 D-PHY Timing Control

The D-PHY timing control is clock-based. All timing parameters defined in D-PHY specification is counted by D-PHY internal clock. In Table. 5-1, it's shown that the D-PHY clock cycle time is related to PLL clock settings and rg_nx_ck_sel. For more precise timing control, the D-PHY clock should be selected as fast as possible, however, the faster the clock, the more power wasted. To select a suitable D-PHY clock is also important for optimizing the system power consumption.

For example, the timing of $T_{HS\text{-}PREPARE}$ is mandatory to be the value between 40ns+4*UI to 85ns +6*UI. In the case of VGA display shown in Table. 5-1, UI is 300 Mhz cycle time. In other words, the value of the timing parameter $T_{HS\text{-}PREPARE}$ must be 53.2 ~ 104.8ns which could be inferred to our register settings as shown in Table. 5-4. Please see section 5.10.8 for the detail of D-PHY timing specification.

Table. 5-4: D-PHY timing parameters register settings

	Timing specification	**Absolute time for UI is 3.3ns**	**Timing settings for rg_nx_ck_sel**		
			0	1	2
$T_{HS\text{-}PREPARE}$	40ns+4*UI ~ 85ns +6*UI	53.2 ns ~ 104.8 ns	17 ~ 31	8 ~ 15	6 ~ 10

5.10.6 Video Mode Operation

MT6516 DSI controller supports all of the DSI video mode traffic sequences including Sync Pulse Mode, Sync Event Mode and Burst Mode. To facilitate the translation of the parameters of the packets, the timing diagrams and corresponding register settings are illustrated below.

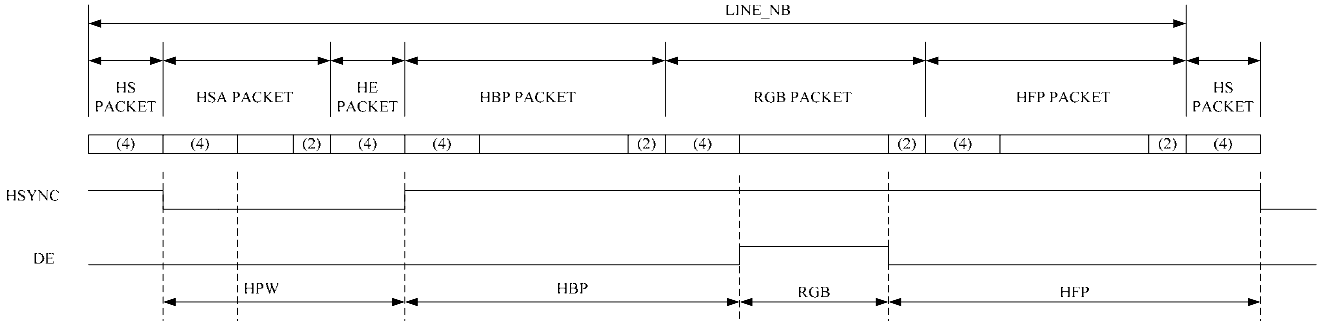


Fig. 5-3: Sync Pulse Mode Timing Diagram

- HSA_NB = HPW*BPP -4
- HBP_NB = HBP*BPP -4
- RGB_NB = RGB*BPP +6
- HFP_NB = HFP*BPP -6
- LINE_NB = 4 + HBP_NB + 4+ RGB_NB + HFP_NB = (HPW + HBP + RGB + HFP)* BPP

Fig. 5-4: Sync Pulse Mode Register Settings

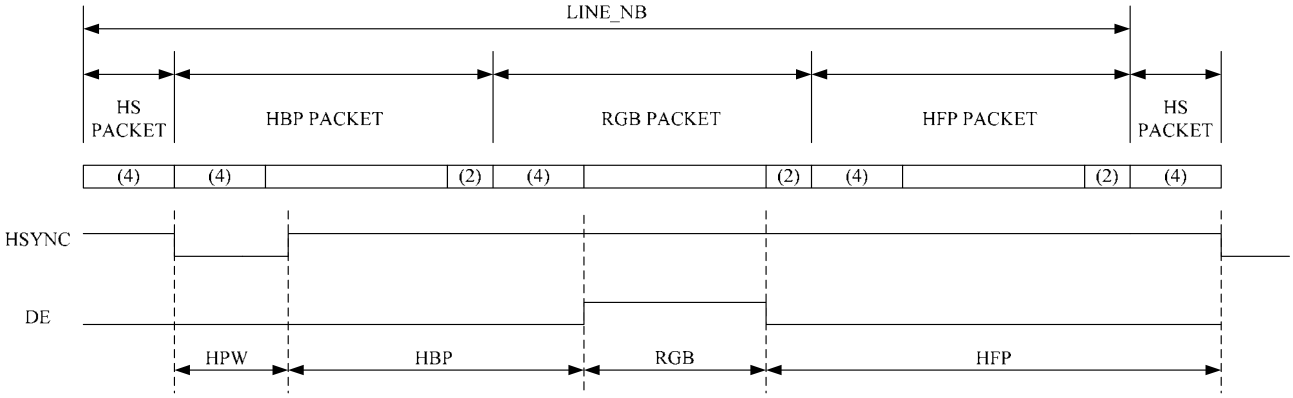


Fig. 5-5: Sync Event Mode Timing Diagram

© 2009 MediaTek Inc.

- HBP_NB = (HPW+HBP)*BPP -4
- RGB_NB = RGB*BPP +6
- HFP_NB = HFP*BPP -6
- LINE_NB = 4 + HBP_NB + RGB_NM + HFP_NB = (HPW + HBP + RGB + HFP)* BPP

Fig. 5-6: Sync Event Mode Register Settings

5.10.7 Command Queue Operation

MT6516 DSI controller has two dedicated command queue which is 32-bit wide and 16-entry depth for each, please refer to Fig. 5-7. To simplify the settings for transmitting a packet in command mode, this command queue is designed to categorize all possible transmission types and commands into four main instructions and unify all the DSI specification commands into one or several 32-bit wide instructions. Fig. 5-7 also illustrates the 32-bit instruction structure where the CONFG byte gives the instruction format.

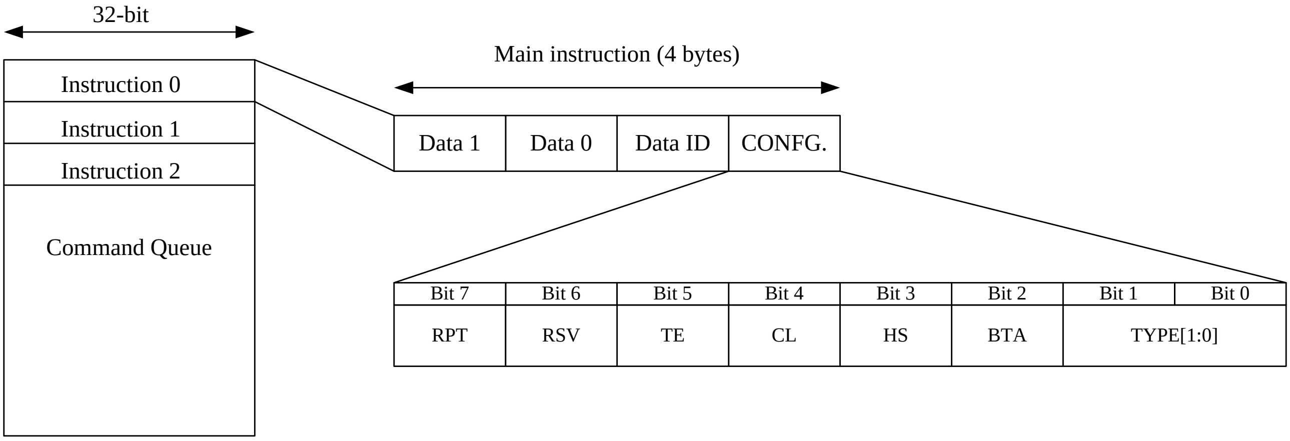


Fig. 5-7: DSI command queue instruction type

Table. 5-5 shows the detail descriptions of the config. field of the main instruction. For more detail, please refer to section 5.10.7.1 to 5.10.7.4. For convenience, we use virtual channel 0 for all packet in all examples of this document. The real virtual channel numbers depends on the definition of slave side which needs to be specially taken care by SW programmer.

Table. 5-5: Config. field description of main instruction

	Value	Function	Description
Type[1:0]	00	-	Used for DSI short packet read/write command
	01	-	Used for DSI frame buffer write command (long packet)
	10	-	Used for DSI generic long packet write command
	11	-	Used for DSI frame buffer read command (short packet)
BTA	0	Off	Turn-around the DSI link after this DSI command is transmitted
	1	On	
HS	0	Off	Enable HS TX transmission for this packet, otherwise transmit packet via LP TX
	1	On	
CL	0	8-bit	Command length selection for frame buffer read/write instruction, only effective for type1 and type3 instructions.
	1	16-bt	
TE	0	On	Enable TE request, which will only turn-around the DSI link without any packet transmission
	1	Off	
Resv	-	-	Reserved for further use

	Value	Function	Description
RPT	0	On	Repeat the payload data “0” WC times, where WC is the packet Word Count defined by DSI specification
	1	Off	

5.10.7.1 Type-0 instruction

Type-0 instruction is used to transmit short packets. Fig. 5-8 shows type-0 instruction format, where (Data ID + Data 0 + Data 1) is constructed by a DSI short packet command (without ECC).

Byte 3	Byte 2	Byte 1	Byte 0
Data 1	Data 0	Data ID	CONFG.

Fig. 5-8: Type-0 instruction format

Suppose that we are going to send “Turn On Peripheral” and “Color Mode On” commands which are transmitted via LP TX and HS TX respectively. In addition, we want to request slave response after the second command finished. These descriptions could be translated into two 32-bit instructions and achieved by the steps shown in Table. 5-6.

Table. 5-6: Type-0 TX example

Step	Description	R/W	Address	Value
I	Fill the command queue entry-0 with value “0x0000_3200”	W	0xC000	0x0000_3200
II	Fill the command queue entry-1 with value “0x0000_120C”	W	0xC004	0x0000_120C
III	Set the command count as 2	W	0x60	0x2
IV	Start command	W	0x0	0x1
V	Interrupt issued, received slave response	R	0xC	0x1
VI	Read status	R	0x70	bit-15 = 0x1
VII	Read trigger status	R	0x88	0x2
VIII	Go to next instruction in DSI command queue	W	0x84	0x1
IX	Interrupt issued, all instructions done	R	0xC	0x2

5.10.7.2 Type-1 instruction

Type-1 command is used to write data into frame buffer. As shown in Fig. 5-9, there are four bytes constructing this type of instruction where Mem_start_0 and Mem_start_1 could be generic commands defined by slave vendors or DCS commands. Mem_start_1 is optional, that is, the memory start/continue command could be single byte like DCS defined. To indicate the DSI controller whether the Mem_start_1 is sent or not depends on the CL bit of the CONFG. byte. Since the length we are going to update the frame buffer is not a constant, this type of instruction may send several long packets to slave side. The payload data and length of each packet (excluding the mem_start_0 and mem_start_1) is prepared by LCD controller which

couples the output of image data path or layer overlay result to DSI controller. For the first packet, mem_start_0 and mem_start_1 (if CL = 1) is used as the parameters to inform slave side that host is starting to write the frame buffer. For the remaining packets, the register value MEM_CONTI[31:16] will be used as the parameters to tell slave side write these data following the last pixel of previous packet. For more flexibility, Mem_start_0, Mem_start_1, MEM_CONTI[31:16] and CL are all programmable, however, it consumes only one entry of command queue.

Byte 3	Byte 2	Byte 1	Byte 0
Mem start 1 (optional)	Mem start 0	Data ID	CONFG.

Fig. 5-9: Type-1 instruction format

Here gives an example to write frame buffer via DCS command in HS TX mode.

Table. 5-7: Type-1 TX example

Step	Description	R/W	Address	Value
I	Fill the command queue entry-0 with value "0x002c3909"	W	0xC000	0x002C_3909
II	Set the command count as 1	W	0x60	0x1
III	Set memory continue bytes	W	0x90	[31:16] = 0x3C
IV	Start command	W	0x0	0x1
V	Interrupt issued, all instructions done	R	0xC	0x2

5.10.7.3 Type-2 instruction

Type-2 instruction is used to send a long packet. As shown in Fig. 5-10, this type of main-instruction requires several sub-instructions which don't have CONFIG. It's obviously that host needs to prepare the content of the packet (excluding ECC and checksum) in the command queue to send a generic long packet. In Fig. 5-10, the byte2 and byte3 will be ignored and the next slot of instruction will be treated as a main-instruction which byte 0 will be parsed as a CONFIG. byte.
According to this type of instruction, it's strongly recommended to send a long packet to slave side in LP TX mode because the command since there is cross clock domain latency on reading the sub-instructions.

Byte 3	Byte 2	Byte 1	Byte 0
WC 1	WC 0	Data ID	CONFG.
Data 3	Data 2	Data 1	Data 0
		Data WC-1	Data WC-2

Fig. 5-10: Type-2 instruction format

For example, we are going to send a "0x112233" using a DSI generic write command. Please refer to Table. 5-8.

Table. 5-8: Type-2 TX example

Step	Description	R/W	Address	Value
I	Fill the command queue entry-0 with value “0x00032902”	W	0xC000	0x0003_2902
II	Fill the command queue entry-1 with value “0x00112233”	W	0xC004	0x0011_2233
III	Set the command count as 2	W	0x60	0x2
IV	Start command	W	0x0	0x1
V	Interrupt issued, all instructions done	R	0xC	0x2

It’s notice that the RPT bit of CONFIG. is designed for this type of instruction. It’s useful for NULL packet or blanking packet. For example, if a null packet is going to be sent, only the main-instruction (entry-0 of command queue) is needed, the following payload data will be sent as “0”.

5.10.7.4 Type-3 instruction

Type-3 instruction is used for frame buffer read. As shown in Fig. 5-11, the format is the same as that of type-1. When executing this instruction, host will firstly send a short packet with memory start parameter given in byte2 and byte3 and automatically issues next packet by memory continue parameters which is programmed in MEM_CONTI[15:0]. The number of total packets required to be sent depends on the FRM_BC and “maximum return packet size”. For example, if we are going to read 1024 bytes from the frame buffer in slave side and the “maximum return packet size” is set as “4”, after the first short packet described in main-instruction is sent, there are another 255 short packets with memory continue parameters needed to be sent successively.

Byte 3	Byte 2	Byte 1	Byte 0
Men start 1 (optional)	Mem start 0	Data ID	CONFG.

Fig. 5-11: Type-3 instruction format

Table. 5-9 gives a example of using type-3 instruction to perform frame buffer read.

Table. 5-9: Type-3 TX example

Step	Description	R/W	Address	Value
I	Fill the command queue entry-0 with value “0x002e0603”	W	0xC000	0x002e_0603
II	Set the command count as 1	W	0x60	0x1
III	Set memory continue bytes	W	0x90	[15:0] = 0x3E
IV	Start command	W	0x0	0x1
V	Interrupt issued, received slave response	R	0xC	0x1
VI	Read status	R	0x70	-
VII	Start next read or go to next instruction	W	0x84	0x1
VIII	Interrupt issued, all instructions done	R	0xC	0x2

 © 2009 MediaTek Inc.

5.10.8 Appendix

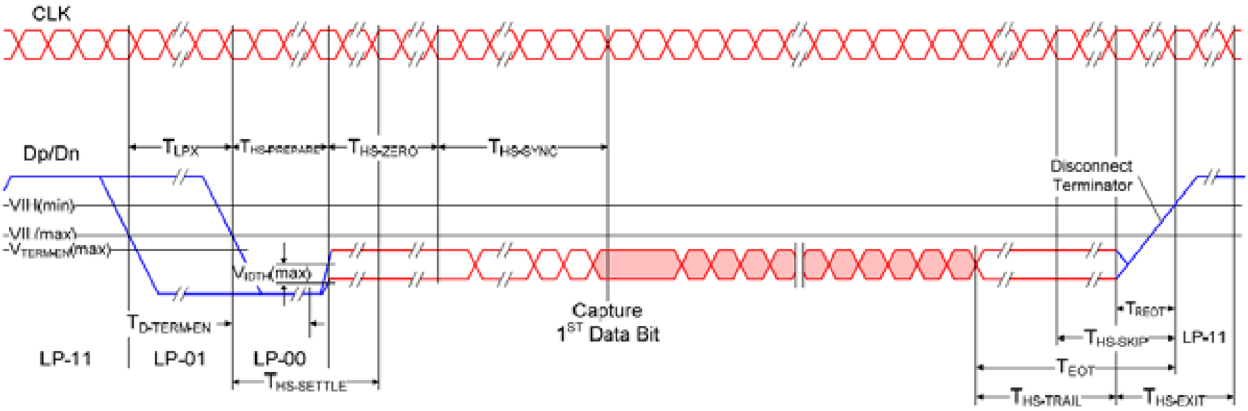


Fig. 5-12: D-PHY HS TX timing specification

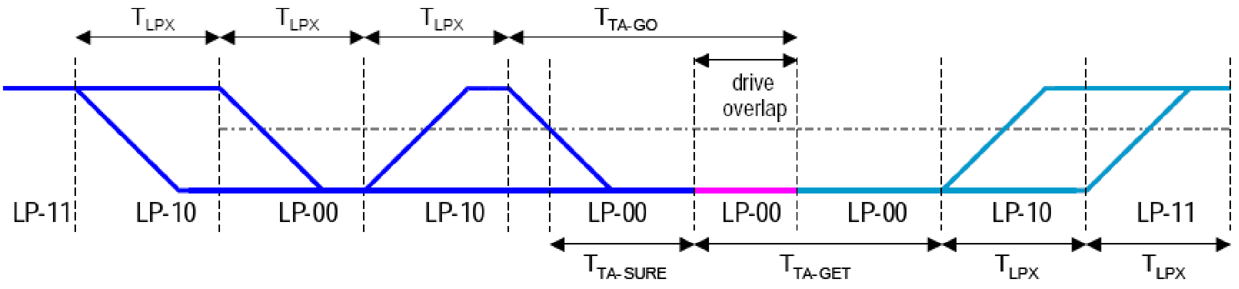


Fig. 5-13: D-PHY BTA timing specification

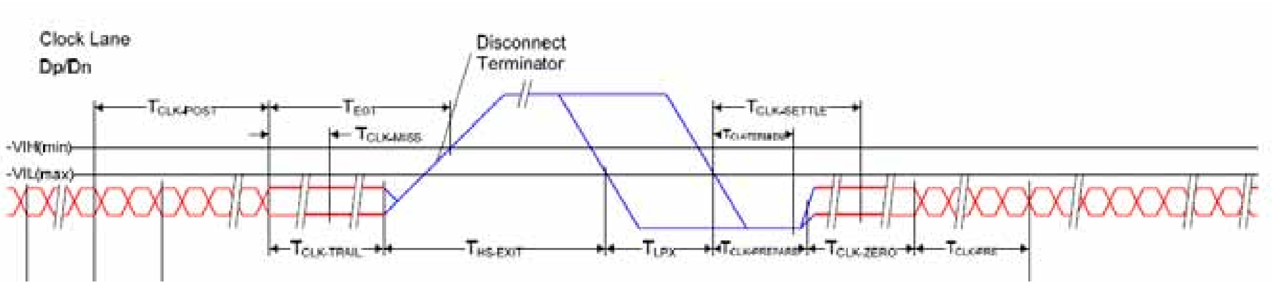


Fig. 5-14: D-PHY clock lane timing specification

Parameter	Description	Min	Typ	Max	Unit	Notes
$T_{CLK-MISS}$	Detection time that the clock has stopped toggling			60	ns	1
$T_{CLK-POST}$	Time that the transmitter shall continue sending HS clock after the last associated Data Lane has transitioned to LP mode	60 ns + 52*UI			ns	
$T_{CLK-PRE}$	Time that the HS clock shall be driven prior to any associated Data Lane beginning the transition from LP to HS mode	8			UI	
$T_{CLK-PREPARE}$	Time to drive LP-00 to prepare for HS clock transmission	38		95	ns	
$T_{CLK-TERM-EN}$	Time to enable Clock Lane receiver line termination measured from when Dn crosses $V_{IL,MAX}$	Time for Dn to reach $V_{TERM-EN}$		38	ns	
$T_{CLK-TRAIL}$	Time to drive HS differential state after last payload clock bit of a HS transmission burst	60			ns	
$T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$	$T_{CLK-PREPARE}$ + time for lead HS-0 drive period before starting Clock	300			ns	
$T_{D-TERM-EN}$	Time to enable Data Lane receiver line termination measured from when Dn crosses $V_{IL,MAX}$.	Time for Dn to reach $V_{TERM-EN}$		35 ns + 4*UI		
T_{EOT}	Time from start of $T_{HS-TRAIL}$ or $T_{CLK-TRAIL}$ period to start of LP-11 state			105 ns + n*12*UI		3
$T_{HS-EXIT}$	Time to drive LP-11 after HS burst	100			ns	
$T_{HS-PREPARE}$	Time to drive LP-00 to prepare for HS transmission	40 ns + 4*UI		85 ns + 6*UI	ns	
$T_{HS-PREPARE}$ + $T_{HS-ZERO}$	$T_{HS-PREPARE}$ + Time to drive HS-0 before the Sync sequence	145 ns + 10*UI			ns	
$T_{HS-SKIP}$	Time-out at RX to ignore transition period of EoT	40		55 ns + 4*UI	ns	
$T_{HS-TRAIL}$	Time to drive flipped differential state after last payload data bit of a HS transmission burst	max(n*8*UI, 60 ns + n*4*UI)			ns	2, 3
T_{INIT}	Initialization period (PHY might calibrate)	100			µs	
T_{LPX}	Length of any Low-Power state period	50			ns	4
Ratio T_{LPX}	Ratio of $T_{LPX(MASTER)}/T_{LPX(SLAVE)}$ between Master and Slave side	2/3		3/2		
T_{TA-GET}	Time to drive LP-00 by new TX		5*T_{LPX}		ns	
T_{TA-GO}	Time to drive LP-00 after Turnaround Request		4*T_{LPX}		ns	
$T_{TA-SURE}$	Time-out before new TX side starts driving	T_{LPX}		2*T_{LPX}	ns	
T_{WAKEUP}	Recovery time from Ultra-Low Power State	1			ms	

Fig. 5-15: D-PHY timing parameters

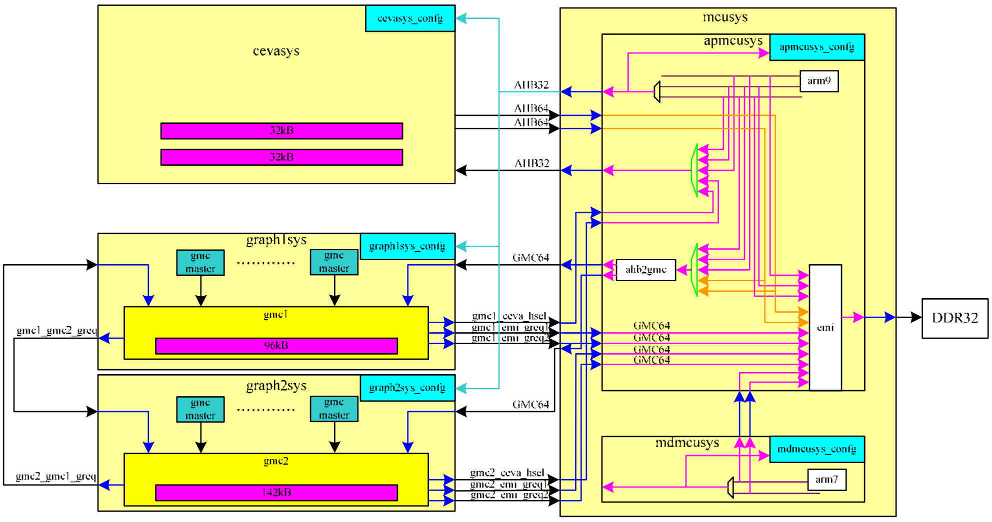
5.11 Graphics Memory Interface

5.11.1 Introduction

For providing graphic engines an easier bus protocol, we develop another protocol named GMC used for all the graph related engines. GMC will dispatch the data transaction command to 3 slavers they are internal SRAM, external memory and CEVA SRAM. Each slaver owns their arbiter, the arbiter using two

layer arbitration one is normal priority the other is ultra high priority. For preventing from error base address access (such as some gmc engine access internal rom), the GMC provide a function to stop this illegal access to wrong base address, it will hang on the port to prevent it from corrupting the other data and report it in the register and issue an interrupt to CPU. The base address for GMC is list as follow:

- 0x0000_0000 128MB External RAM
- 0x1000_0000 128MB External RAM
- 0x2000_0000 128MB External RAM
- 0x3000_0000 128MB External RAM
- 0x4000_0000 ~ 0x4001_7fff 96 KB Internal GMC1 SRAM
- 0x4002_0000 ~ 0x4004_3fff 144KB Internal GMC2 SRAM
- 0xB020_0000 ~ 0xB020_7fff 32KB SRAM CEVA
- 0xB040_0000 ~ 0xB040_7fff 32KB SRAM CEVA



The 36 master engines on GMC1 is list as follows:

[0] tvc_pfh_rd
[1] dpi
[2] defect (CAM)
[3] imgdma0
[4] ap_gmc1
[5] lcd_r
[6] lcd_w
[7] tvc
[8] asm
[9] cam
[10] g1fake

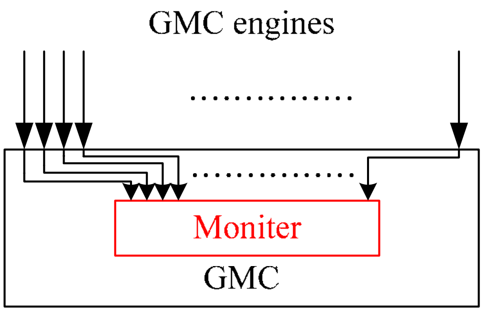
[11] tvc_pfh_wr
[12] dpi_hwc
[13] gmc2_gmc1
[14] imgdma1
[15] imgdma2
[16] imgdma3
[17] imgdma4
[18] g2d_rd
[19] spi
[20] wave
[21] g2d_wr
[22] gcmq
[23] png1
[24] png2
[25] png3
[26:35] NOT USED
The 42 master engines on GMC2 is list as follows:

[39] reserved
[38] reserved
[37] mp4_deblk_0
[36] mp4_deblk_1
[35] fake_eng
[34] imgdma1_1
[33] imgdma1_2
[32] imgdma1_3
[31] prz_blkr
[30] prz_blkw
[29] prz_pxl
[28] m3d_index
[27] m3d_vc
[26] m3d_mvtx
[25] m3d_hvtx
[24] m3d_ez
[23] m3d_txc
[22] m3e_cc_wb
[21] m3d_cc_lf
[20] m3d_zc_wb
[19] m3d_zc_lf
[18] m3d_sc_lf
[17] mux[0] (mp4 or h264)
[16] jpg_dec0
[15] jpg_dec1

 © 2009 MediaTek Inc.

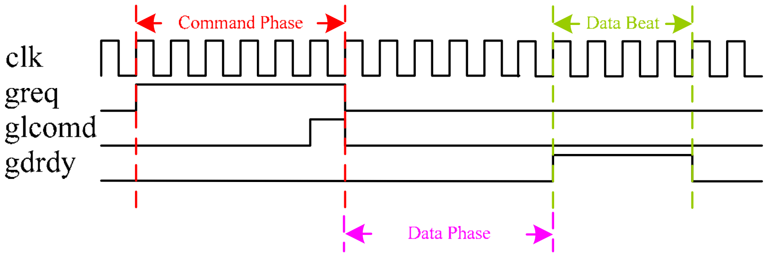
[14:4] mux[11:1] (mp4 or h264)
[3] jpg_enc
[2] imgdma1
[1] apmcu_gmc2
[0] gmc1_gmc2

5.11.2 Performance Counter

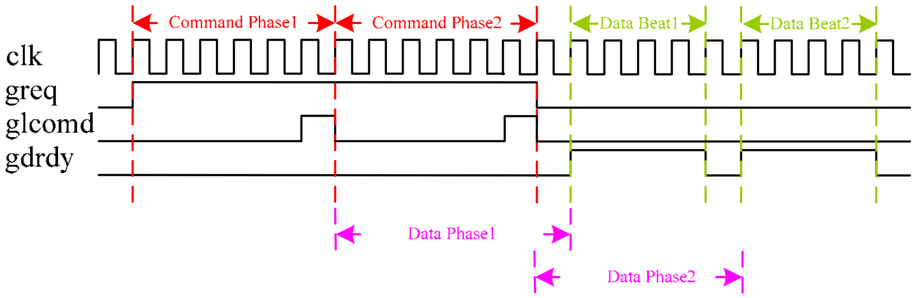


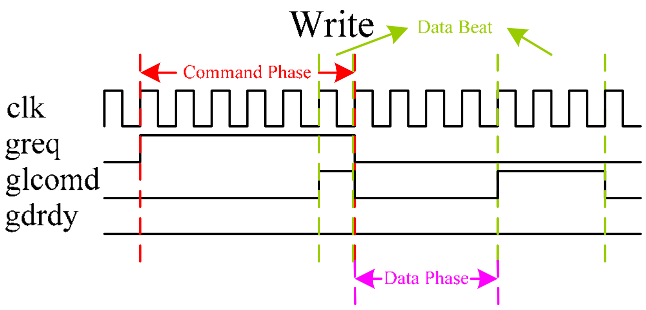
Active window
Idle Cycle

Single Read



Multiple Read





5.11.3 GMC1 Register Definitions

45. GMC1 Arbitration Mode

GMC1 + 0000h GMC1_ARB_MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ARB_MODE**
Type																**R/W**
Reset																**0**

Comment:

1: allow normal to join arbitration

0: only ultra high can join to arbitrate

46. GMC1 Interrupt Enable

GMC1 + 0004h GMC1_INT_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**INT_EN**
Type																**R/W**
Reset																**0**

Comment:

1: allow GMC issue interrupt

0: not allow GMC issue interrupt

47. GMC1 Interrupt Flag1

GMC1 + 0008h GMC1_INT_FLAG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**P31**	**P30**	**P29**	**P28**	**P27**	**P26**	**P25**	**P24**	**P23**	**P22**	**P21**	**P20**	**P19**	**P18**	**P17**	**P16**
Type	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Comment:

Specify the error happened in which ports.

1: error

0: no

48. GMC1 Interrupt Flag2

GMC1 + 000Ch **GMC1_INT_FLAG2**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													P35	P34	P33	P32
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

Comment:

Specify the error happened in which ports.

49. GMC1 Performance Counter Enable

GMC1 + 0010h **GMC1_PCNT_EN**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_EN
Type																R/W
Reset																0

Comment:

1: start to count

0: stop counting

50. GMC1 Performance Counter Clear

GMC1 + 0014h **GMC1_PCNT_CLR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_CLR

Type																**R/W**
Reset																**0**

Comment:

1: start to clear

0: stop clearing

51. GMC1 Performance Counter Selector

GMC1 + 0018h GMC1_ PCNT_PSEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**PCNT_PSEL**					
Type											**R/W**					
Reset											**0**					

Comment:

psel == 0 will choose port[0] and so on

52. GMC1 Performance Counter R/W Selector

GMC1 + 001Ch GMC1_ PCNT_RWSEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**PCNT_RWSEL**		
Type														**R/W**		
Reset														**0**		

Comment:

3’b000: count internal and external R and W transaction

3’b001: count external R and W

3’b010: count internal R and W

3’b011: count external R only

3’b100: count external W only

3’b101: count internal R only

3’b110: count internal W only

53. GMC1 Performance Counter Idle Selector

GMC1 + 0020h GMC1_PCNT_ISEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_ISEL
Type																R/W
Reset																0

Comment:

1: read data phase don't count in idle if no greq

0: read data phase will count in idle if no greq

54. GMC1 Performance Counter Max Selector

GMC1 + 0024h GMC1_PCNT_MAX _SEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_MAX_ SEL
Type																R/W
Reset																0

Comment:

1: count command phase max

0: count data phase max

55. GMC1 Performance Counter Active Count

GMC1 + 0028h GMC1_ PCNT_ACNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_ACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_ACNT															
Type	R/W															
Reset	0															

Comment:

PCNT_ACNT will start to count how many cycles when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

56. GMC1 Performance Counter Idle Count

GMC1 + 002Ch GMC1_ PCNT_ICNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_ICNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	PCNT_ICNT
Type	R/W
Reset	0

Comment:

PCNT_ICNT will count the cycles when pcnt_en = 1 and no any bus transaction is in active, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

57. GMC1 Performance Counter GREQ Count

GMC1 + 0030h **GMC1_ PCNT_GCNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_GCNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_GCNT															
Type	R/W															
Reset	0															

Comment:

GCNT will start to count how many times greq has been issued when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

58. GMC1 Performance Counter GULTRA Count

GMC1 + 0034h **GMC1_ PCNT_UCNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_UCNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_UCNT															
Type	R/W															
Reset	0															

Comment:

GCNT will start to count how many times gultra has been issued when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

59. GMC1 Performance Counter Command Phase Cycle Accumulation Count

GMC1 + 0038h **GMC1_ PCNT_CPCACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_CPCACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_CPCACNT															
Type	R/W															
Reset	0															

Comment:

 © 2009 MediaTek Inc.

CCACNT will start to accumulate the cycles from greq=1 to this command phase end when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

60. GMC1 Performance Counter Data Phase Cycle Accumulation Count

GMC1 + 003Ch **GMC1_ PCNT_DPCACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DPCACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_DPCACNT															
Type	R/W															
Reset	0															

Comment:

DCACNT will start to accumulate the cycles from the start of the data phase to its end when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

61. GMC1 Performance Counter Data Beats Accumulation Count

GMC1 + 0040h **GMC1_ PCNT_DBEACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DBEACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_DBEACNT															
Type	R/W															
Reset	0															

Comment:

DBEACNT will start to accumulate the beat number of each transaction when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

62. GMC1 Performance Counter Data Bytes Accumulation Count

GMC1 + 0044h **GMC1_ PCNT_DBYACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DBYACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_DBYACNT															
Type	R/W															
Reset	0															

Comment:

DBYACNT will start to accumulate the byte number of each transaction when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

63. GMC1 Performance Counter Hang Maximum

GMC1 + 0048h GMC1_ PCNT_HANG_MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**PCNT_HANG_MAX**			
Type													**R/W**			
Reset													**0**			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PCNT_HANG_MAX**															
Type	**R/W**															
Reset	**0**															

Comment:

HANG_MAX will record the max latency cycles for each transaction when pcnt_en = 1, it will be cleared when pcnt_clr = 1

64. GMC1 Performance Counter Maximum

GMC1 + 004Ch GMC1_ PCNT_ MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**PCNT_ MAX**			
Type													**R/W**			
Reset													**0**			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PCNT_ MAX**															
Type	**R/W**															
Reset	**0**															

Comment:

PCNT_MAX will record the max cycles for command phase or data phase determined by psel of each transaction when pcnt_en = 1, it will be cleared when pcnt_clr = 1

65. GMC1 Performance Counter Latch

GMC1 + 0050h GMC1_PCNT_LATCH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**PCNT_LATCH**
Type																**W**
Reset																**0**

Comment:

1: latch the counter value to register

66. GMC1 Performance Counter Latch Clear

GMC1 + 0054h GMC1_PCNT_LATCH_CLEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**PCNT_LATCH_CLR**
Type																**R/W**
Reset																**0**

Comment:

1: latch and no clear, 0: latch and clear

67. GMC1 GMC Direction

GMC1 + 0058h **GMC1_GMC_DIR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**GMC_DIR**
Type																**R/W**
Reset																**0**

Comment:

1: allow GMC2 -> GMC1

0: allow GMC1 -> GMC2

5.11.4 GMC2 Register Definitions

1. GMC2 Arbitration Mode

GMC2 + 0000h **GMC2_ARB_MODE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ARB_MODE**
Type																**R/W**
Reset																**0**

Comment:

1: allow normal to join arbitration

0: only ultra high can join to arbitrate

2. GMC2 Interrupt Enable

GMC2 + 0004h **GMC2_INT_EN**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name																INT_EN
Type																R/W
Reset																0

Comment:

1: allow GMC issue interrupt

0: not allow GMC issue interrupt

3. GMC2 Interrupt Flag1

GMC2 + 0008h **GMC2_INT_FLAG1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Comment:

Specify the error happened in which ports.

1: error 0: no

4. GMC2 Interrupt Flag2

GMC2 + 000Ch **GMC2_INT_FLAG2**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							P41	P40	P39	P38	P37	P36	P35	P34	P33	P32
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset							0	0	0	0	0	0	0	0	0	0

Comment:

Specify the error happened in which ports.

5. GMC2 Performance Counter Enable

GMC2 + 0010h **GMC2_PCNT_EN**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_EN
Type																R/W
Reset																0

Comment:

1: start to count

0: stop counting

6. GMC2 Performance Counter Clear

GMC2 + 0014h **GMC2_PCNT_CLR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_CLR
Type																R/W
Reset																0

Comment:

1: start to clear

0: stop clearing

7. GMC2 Performance Counter Selector

GMC2 + 0018h **GMC2_ PCNT_PSEL**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PCNT_PSEL					
Type											R/W					
Reset											0					

Comment:

psel == 0 will choose port[0] and so on

8. GMC2 Performance Counter R/W Selector

GMC2 + 001Ch **GMC2_ PCNT_RWSEL**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PCNT_RWSEL		
Type														R/W		
Reset														0		

Comment:

3’b000: count internal and external R and W transaction

3’b001: count external R and W

3’b010: count internal R and W

3’b011: count external R only

3'b100: count external W only

3'b101: count internal R only

3'b110: count internal W only

9. GMC2 Performance Counter Idle Selector

GMC2 + 0020h **GMC2_PCNT_ISEL**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_ISEL
Type																R/W
Reset																0

Comment:

1: read data phase don't count in idle if no greq

0: read data phase will count in idle if no greq

10. GMC2 Performance Counter Max/Min Selector

GMC2 + 0024h **GMC2_PCNT_MAX _SEL**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_MAX_SEL
Type																R/W
Reset																0

Comment:

1: count command phase max

0: count data phase max

11. GMC2 Performance Counter Active Count

GMC2 + 0028h **GMC2_ PCNT_ACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_ACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_ACNT															
Type	R/W															
Reset	0															

Comment:

PCNT_ACNT will start to count how many cycles when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

12. GMC2 Performance Counter Idle Count

GMC2 + 002Ch **GMC2_ PCNT_ICNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_ICNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_ICNT															
Type	R/W															
Reset	0															

Comment:

PCNT_ICNT will count the cycles when pcnt_en = 1 and no any bus transaction is in active, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

13. GMC2 Performance Counter GREQ Count

GMC2 + 0030h **GMC2_ PCNT_GCNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_GCNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_GCNT															
Type	R/W															
Reset	0															

Comment:

GCNT will start to count how many times greq has been issued when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

14. GMC2 Performance Counter GULTRA Count

GMC2 + 0034h **GMC2_ PCNT_UCNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_UCNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_UCNT															
Type	R/W															
Reset	0															

Comment:

GCNT will start to count how many times gultra has been issued when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

15. GMC2 Performance Counter Command Phase Cycle Accumulation Count

GMC2 + 0038h **GMC2_ PCNT_CPCACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_CPCACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_CPCACNT															
Type	R/W															
Reset	0															

Comment:

CCACNT will start to accumulate the cycles from greq=1 to this command phase end when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

16. GMC2 Performance Counter Data Phase Cycle Accumulation Count

GMC2 + 003Ch **GMC2_ PCNT_DPCACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DPCACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_DPCACNT															
Type	R/W															
Reset	0															

Comment:

DCACNT will start to accumulate the cycles from the start of the data phase to its end when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

17. GMC2 Performance Counter Data Beats Accumulation Count

GMC2 + 0040h **GMC2_ PCNT_DBEACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DBEACNT															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCNT_DBEACNT															
Type	R/W															
Reset	0															

Comment:

DBEACNT will start to accumulate the beat number of each transaction when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

18. GMC2 Performance Counter Data Bytes Accumulation Count

GMC2 + 0044h **GMC2_ PCNT_DBYACNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCNT_DBYACNT															
Type	R/W															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PCNT_DBYACNT**															
Type	**R/W**															
Reset	**0**															

Comment:

DBYACNT will start to accumulate the byte number of each transaction when pcnt_en = 1, it will saturate when reach 32'hffffffff, and it will be cleared when pcnt_clr = 1

19. GMC2 Performance Counter Hang Maximum

GMC2 + 0048h **GMC2_ PCNT_HANG_MAX**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**PCNT_HANG_MAX**			
Type													**R/W**			
Reset													**0**			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PCNT_HANG_MAX**															
Type	**R/W**															
Reset	**0**															

Comment:

HANG_MAX will record the max latency cycles for each transaction when pcnt_en = 1, it will be cleared when pcnt_clr = 1

20. GMC2 Performance Counter Maximum

GMC2 + 004Ch **GMC2_ PCNT_ MAX**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**PCNT_ MAX**			
Type													**R/W**			
Reset													**0**			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PCNT_ MAX**															
Type	**R/W**															
Reset	**0**															

Comment:

PCNT_MAX will record the max cycles for command phase or data phase determined by psel of each transaction when pcnt_en = 1, it will be cleared when pcnt_clr = 1

21. GMC2 Performance Counter Latch

GMC2 + 0050h **GMC2_PCNT_LATCH**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**PCNT_LATCH**
Type																**W**

Reset																0

Comment:

1: latch the counter value to register

22. GMC2 Performance Counter Latch Clear

GMC2 + 0054h GMC2_PCNT_LATCH_CLEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																PCNT_LATCH_CLR
Type																R/W
Reset																0

Comment:

1: latch and no clear, 0: latch and clear

23. GMC2 MUX PORT SEL

GMC2 + 0058h GMC2_MUX_PORT_SEL

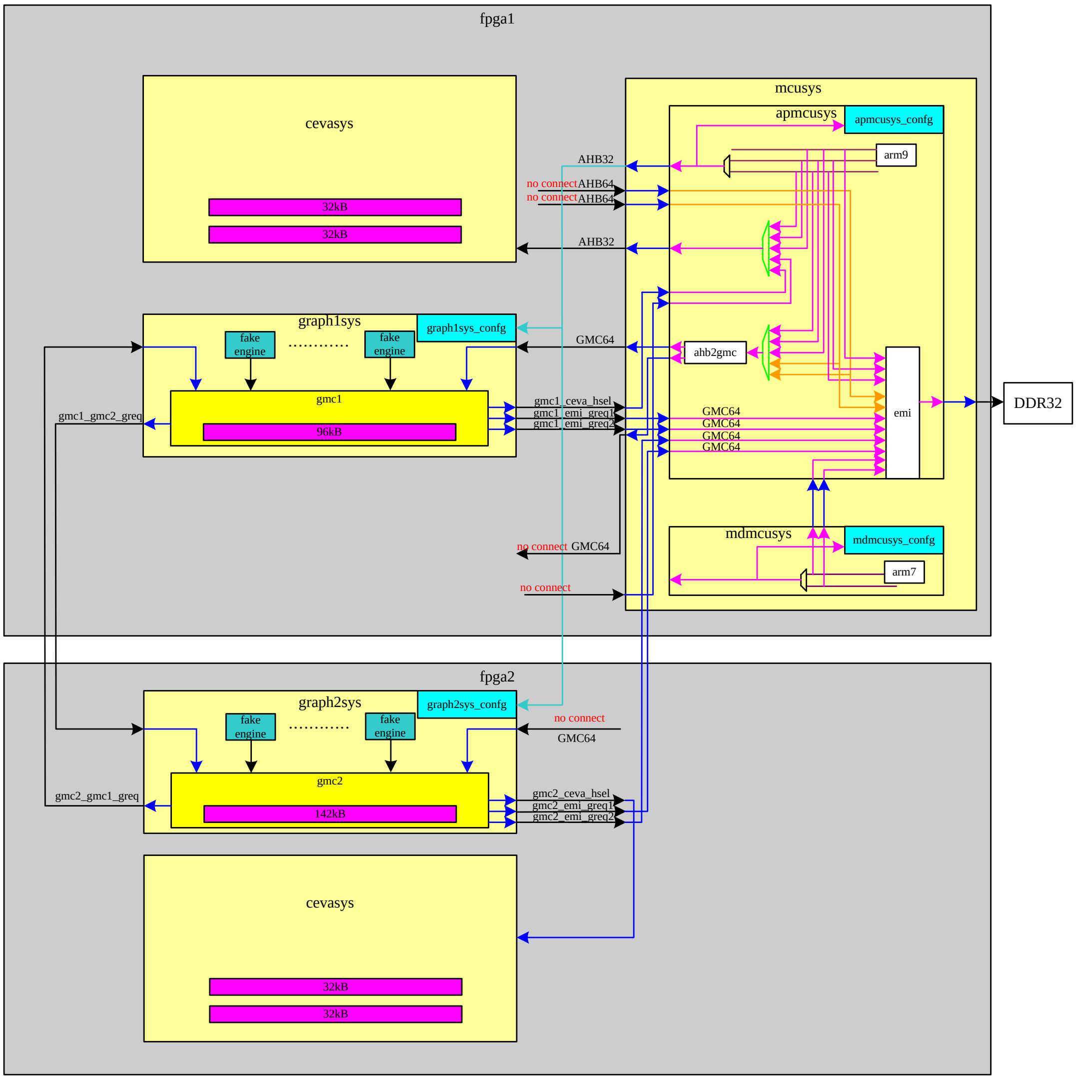
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																MUX_PORT_SEL
Type																R/W
Reset																0

Comment:

5.12 GMC Fake Engine

5.12.1 Introduction

Because of lack of GMC master engines, the FPGA verification for GMC is using fake engine to simulate the GMC behavior. It will issue all kinds of burst and size type plus 8 kinds of address to test the GMC design. First the engine will write a set of data to the GMC and then read it back and compare it. If the data is correct, the engine will issue next data till the whole kinds of data is issued and loop again. Software can specify the loop count to order fake engine to loop. Following is the FPGA diagram.



5.12.2 Fake Engine

Fake engine is a synthesizable RTL hardware design. It is used on FPGA to test GMC engine and the connection between GMC. It will start to work when software enables the "act_en". It will auto issue all kinds of request to access Int./Ext./CEVA memory, and auto compare the data correctness. Software need to configure the loop number for engine to loop, the number at least larger or equal to 1. The ultra signal is used to test weather the ultra high is take effect, if the bit is turn on fake engine will always issue ultra high request to GMC engine and GMC engine will serve it to higher priority and result in fewer active count. However for hardware limitation we only implement 4 ultra high port on GMC engine so only fake engine 26~29 for gmc1 and engine 37~40 for gmc2 the bit will take effect. Software need to clear the fake engine before each start. There will 3 situation the fake will occur when start, error, hang, done. Only done is the correct situation. No matter what happen the act_en will be turn off, software just need to polling this bit and then to see what happen and go to the next test. Each engine will occupy 128 bytes address range and no two engine can

overlap, software need to configure the eng_offset to separate the engine’s address include GMC1 and GMC2. Each engine will issue following type of transaction.

Burst type: 1, 2, 4, 8, 16 = five kinds
Size type: 1, 2, 4, 8 = four kinds
Address[1:0] type: 0~7 = eight kinds
Base address type: = eight kinds

0x0040_0000
0x1040_0000
0x2040_0000
0x3040_0000
0x4000_0000
0x4002_0000
0xB020_0000
0xB040_0000

Base address can be programmed to decide weather to run, because FPGA may not have the corresponding device for that address such as 0x3000_0000. Each engine will issue (5 x 4 x 8 x enabled external address) transactions for each loop.

The max number of transaction for a loop is 5x4x8x8 = 1280. Software need to avoid placing your code on above address region, otherwise fake engine will corrupt the code. The engine will issue address start from base_addr + eng_offset. As we know that the longest data size is 16*8=128 bytes, each engine will occupy such size and none can be overlapped.

5.12.3 Fake Engine Register Base Address

GMC1:

engine00: 0x8009_f000
engine01: 0x8009_e000
engine02: 0x8009_d000
engine03: 0x8009_c000
engine04: 0x8009_b000
engine05: 0x8009_a000
engine06: 0x8009_9000
engine07: 0x8009_8000
engine08: 0x8009_7000
engine09: 0x8009_6000
engine10: 0x8009_5000
engine11: 0x8009_4000
engine12: 0x8009_3000
engine13: 0x8009_2000
engine14: 0x8009_1000
engine15: 0x8009_0000
engine16: 0x8008_f000
engine17: 0x8008_e000
engine18: 0x8008_d000

engine19: 0x8008_c000
engine20: 0x8008_b000
engine21: 0x8008_a000
engine22: 0x8008_9000
engine23: 0x8008_8000
engine24: 0x8008_7000
engine25: 0x8008_6000
engine26: 0x8008_5000
engine27: 0x8008_4000
engine28: 0x8008_3000
engine29: 0x8008_2000
gmc1 : 0x8008_1000
global1 : 0x8008_0000

GMC2:
engine00: 0x800c_a000
engine01: 0x800c_9000
engine02: 0x800c_8000
engine03: 0x800c_7000
engine04: 0x800c_6000
engine05: 0x800c_5000
engine06: 0x800c_4000
engine07: 0x800c_3000
engine08: 0x800c_2000
engine09: 0x800c_1000
engine10: 0x800c_0000
engine11: 0x800b_f000
engine12: 0x800b_e000
engine13: 0x800b_d000
engine14: 0x800b_c000
engine15: 0x800b_b000
engine16: 0x800b_a000
engine17: 0x800b_9000
engine18: 0x800b_8000
engine19: 0x800b_7000
engine20: 0x800b_6000
engine21: 0x800b_5000
engine22: 0x800b_4000
engine23: 0x800b_3000
engine24: 0x800b_2000
engine25: 0x800b_1000
engine26: 0x800b_0000

engine27: 0x800a_f000
engine28: 0x800a_e000
engine29: 0x800a_d000
engine30: 0x800a_c000
engine31: 0x800a_b000
engine32: 0x800a_a000
engine33: 0x800a_9000
engine34: 0x800a_8000
engine35: 0x800a_7000
engine36: 0x800a_6000
engine37: 0x800a_5000
engine38: 0x800a_4000
engine39: 0x800a_3000
engine40: 0x800a_2000
gmc2 : 0x800a_1000
global2 : 0x800a_0000

The number of fake engine for gmc1 is 30
The number of fake engine for gmc2 is 41
The ultra high engine for gmc1 is engin29, engine28, engine27, engine26
The ultra high engine for gmc2 is engin40, engine39, engine38, engine37
The base address of global1 is used to set the act_en for fake engines on gmc1
The base address of global2 is used to set the act_en for fake engines on gmc2

5.12.4 Register Definitions

68. Engine Offset

FAKE_ENG + 0000h **FAKE_ENG_ENG_OFFSET**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENG_OFFSET[15:0]															
Type	R/W															
Reset	0															

Comment:

This offset is used to separate each fake engine's occupied address. Each engine will occupy 128 bytes that is the minimum number of the offset is 128 bytes. Software should very care that no one engine can overlap in address otherwise the data may be corrupted by other engine and result in compare fail.

Ex.

Eng1: offset = 0x0

Eng2: offset = 0x80

Eng3: offset = 0x100

 © 2009 MediaTek Inc.

In FPGA1 and FPGA2 the offset is also need to be separated.

69. Engine Loop Number

FAKE_ENG + 0004h FAKE_ENG_LOOP_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									LOOP_NUM[7:0]							
Type									R/W							
Reset									0							

Comment:

Specify the loop number for each fake engine. A reasonable number is 10. it can't be 0 when turn on fake engine.

70. Engine ULTRA

FAKE_ENG + 0008h FAKE_ENG_ULTRA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ULTRA
Type																R/W
Reset																0

Comment:

It specify weather this engine issue ultra high signal to require more bandwidth. This bit is only effective on engine0 ~ engine3. This bit is used to test the ultra high function. When turn on the ultra high, the number of active counter would be dramatically decreased.

71. Engine Active Enable

FAKE_ENG + 000Ch FAKE_ENG_ACT_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ACT_EN
Type																R/W
Reset																0

Comment:

act_en = 1,start the fake engine. act_en= 0, stop the fake engine

72. Engine Clear Enable

FAKE_ENG + 0010h FAKE_ENG_CLR_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**CLR_EN**
Type																**R/W**
Reset																**0**

Comment:

clr_en = 1, reset the fake engine. clr_en= 0, normal mode.

Remember to reset the fake engine before normal function.

73. Engine Error

FAKE_ENG + 0014h FAKE_ENG_ERR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**ERR**
Type																**R/W**
Reset																**0**

Comment:

err=1, data compare error. err=0, data compare ok.

74. Engine Done

FAKE_ENG + 0018h FAKE_ENG_DONE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**DONE**
Type																**R/W**
Reset																**0**

Comment:

done=1, fake engine finish all the loop. done=0, have not finish.

75. Engine Hang

FAKE_ENG + 001Ch FAKE_ENG_HANG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name															HANG
Type															R/W
Reset															0

Comment:

hang=1, fake engine is hang over 100000 cycles hang=0, fake engine is not hang

76. Engine Active Counter

FAKE_ENG + 0020h FAKE_ENG_ACT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ACT_CNT[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ACT_CNT[15:0]															
Type	R/W															
Reset	0															

Comment:

A counter counts when act_en == 1.

77. Engine Hang Counter

FAKE_ENG + 0024h FAKE_ENG_HANG_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HANG_CNT[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HANG_CNT[15:0]															
Type	R/W															
Reset	0															

Comment:

A counter counts when greq == 1 and reset when data transfer complete.

78. Engine GREQ

FAKE_ENG + 0028h FAKE_ENG_GREQ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																GREQ
Type																R
Reset																0

Comment:

For debug.

79. Engine GADDR

FAKE_ENG + 002Ch **FAKE_ENG_GADDR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GADDR[31:16]															
Type	R															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GADDR[15:0]															
Type	R															
Reset	0															

Comment:

For debug.

80. Engine GBURST

FAKE_ENG + 0030h **FAKE_ENG_GBURST**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														GBURST		
Type														R		
Reset														0		

Comment:

For debug.

81. Engine GSIZE

FAKE_ENG + 0034h **FAKE_ENG_GSIZE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															GSIZE	
Type															R	
Reset															0	

Comment:

For debug.

82. Engine GWRITE

FAKE_ENG + 0038h **FAKE_ENG_GWRITE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																GWRITE
Type																R

Reset																0

Comment:

For debug.

83. Engine GWDATA1

FAKE_ENG + 003Ch FAKE_ENG_GWDATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GWDATA1[31:16]															
Type	R															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GWDATA1[15:0]															
Type	R															
Reset	0															

Comment:

For debug.

84. Engine GWDATA0

FAKE_ENG + 0040h FAKE_ENG_GWDATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GWDATA0[31:16]															
Type	R															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GWDATA0[15:0]															
Type	R															
Reset	0															

Comment:

For debug.

85. Engine Loop Counter

FAKE_ENG + 0044h FAKE_ENG_LOOP_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									LOOP_CNT[7:0]							
Type									R							
Reset									0							

Comment:

Current engine loop count.

86. Engine Base Counter

FAKE_ENG + 0048h FAKE_ENG_BASE_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**BASE_CNT**		
Type														**R**		
Reset														**0**		

Comment:

For debug.

87. Engine Burst Counter

FAKE_ENG + 004Ch FAKE_ENG_BURST_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**BURST_CNT**		
Type														**R**		
Reset														**0**		

Comment:

For debug.

88. Engine Size Counter

FAKE_ENG + 0050h FAKE_ENG_SIZE_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**SIZE_CNT**	
Type															**R**	
Reset															**0**	

Comment:

For debug.

89. Engine Address Counter

FAKE_ENG + 0054h FAKE_ENG_ADDR_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**ADDR_CNT**		
Type														**R**		
Reset														**0**		

Comment:

For debug.

90. Engine Write Counter

FAKE_ENG + 0058h **FAKE_ENG_W_CNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													W_CNT			
Type													R			
Reset													0			

Comment:

For debug.

91. Engine Read Counter

FAKE_ENG + 005Ch **FAKE_ENG_R_CNT**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													R_CNT			
Type													R			
Reset													0			

Comment:

For debug.

92. Base Address Enable

FAKE_ENG + 0060h **FAKE_ENG_BASE_ADDR_EN**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BASE_ADDR_EN[7:0]							
Type									R/W							
Reset									0							

Comment:

Enable for external memory address. 1: turn on, 0: turn off

BASE_ADDR_EN[0]: 0x0040_0000 BASE_ ADDR_EN[1]: 0x1040_0000

BASE_ ADDR_EN[2]: 0x2040_0000 BASE_ ADDR_EN[3]: 0x3040_0000

BASE_ ADDR_EN[4]: 0x4000_0000 BASE_ ADDR_EN[5]: 0x4002_0000

BASE_ ADDR_EN[6]: 0xb020_0000 BASE_ ADDR_EN[7]: 0xb040_0000

When act_en = 1, at least one should be turn on.

93. Engine Hang Counter Maximum

FAKE_ENG + 0064h **FAKE_ENG_HANG_CNT_MAX**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HANG_CNT_MAX[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HANG_CNT_MAX[15:0]															
Type	R/W															
Reset	0															

Comment:

Will preserve the maximum number of hang counter

94. Same Mode

FAKE_ENG + 0068h **FAKE_ENG_SAME_MODE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SAME_MODE
Type																R
Reset																0

Comment:

1'b0: auto check mode 1'b1: the same type mode

95. Read Write Mode

FAKE_ENG + 006Ch **FAKE_ENG_RW_MODE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															RW_MODE	
Type															R	
Reset															0	

Comment:

2'b00 R-R 2'b01 R-W 2'b10 W-R 2'b11 W-W

96. Request Mode

FAKE_ENG + 0070h **FAKE_ENG_REQ_MODE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name											**REQ_MODE**
Type											**R/W**
Reset											**0**

Comment:

5'h0 : burst = 1 beats, size = 1 bytes 5'h1 : burst = 1 beats, size = 2 bytes 5'h2 : burst = 1 beats, size = 4 bytes

5'h3 : burst = 1 beats, size = 8 bytes 5'h4 : burst = 2 beats, size = 1 bytes 5'h5 : burst = 2 beats, size = 2 bytes

5'h6 : burst = 2 beats, size = 4 bytes 5'h7 : burst = 2 beats, size = 8 bytes 5'h8 : burst = 4 beats, size = 1 bytes

5'h9 : burst = 4 beats, size = 2 bytes 5'ha : burst = 4 beats, size = 4 bytes 5'hb : burst = 4 beats, size = 8 bytes

5'hc : burst = 8 beats, size = 1 bytes 5'hd : burst = 8 beats, size = 2 bytes 5'he : burst = 8 beats, size = 4 bytes

5'hf : burst = 8 beats, size = 8 bytes 5'h10: burst = 16 beats, size = 1 bytes 5'h11: burst = 16 beats, size = 2 bytes

5'h12: burst = 16 beats, size = 4 bytes 5'h13: burst = 16 beats, size = 8 bytes

Others: burst = 16 beats, size = 8 bytes

97. Engine Active Enable0 (Global1)

GLOBAL1+ 0000h FAKE_ENG_ACT_EN0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**EN29**	**EN28**	**EN27**	**EN26**	**EN25**	**EN24**	**EN23**	**EN22**	**EN21**	**EN20**	**EN19**	**EN18**	**EN17**	**EN16**
Type			**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**
Reset			**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN15**	**EN14**	**EN13**	**EN12**	**EN11**	**EN10**	**EN09**	**EN08**	**EN07**	**EN06**	**EN05**	**EN04**	**EN03**	**EN02**	**EN01**	**EN00**
Type	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**
Reset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Comment:

act_en = 1, turn of fake engine and it will start to send gmc request.

act_en= 0, fake engine finish the job.

Software keep polling this bit to see if the fake engine is done its job no matter error occurred or GMC was hanged this bit will be turn off.

98. Engine Active Enable0 (Global2)

GLOBAL2+ 0000h FAKE_ENG_ACT_EN0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EN31**	**EN30**	**EN29**	**EN28**	**EN27**	**EN26**	**EN25**	**EN24**	**EN23**	**EN22**	**EN21**	**EN20**	**EN19**	**EN18**	**EN17**	**EN16**
Type	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**
Reset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN15**	**EN14**	**EN13**	**EN12**	**EN11**	**EN10**	**EN09**	**EN08**	**EN07**	**EN06**	**EN05**	**EN04**	**EN03**	**EN02**	**EN01**	**EN00**
Type	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**
Reset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

99. Engine Active Enable1 (Global2)

GLOBAL2+ 0004h FAKE_ENG_ACT_EN1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								EN40	EN39	EN38	EN37	EN36	EN35	EN34	EN33	EN32
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

Comment:

act_en = 1, turn of fake engine and it will start to send gmc request.

act_en= 0, fake engine finish the job.

Software keep polling this bit to see if the fake engine is done its job no matter error occurred or GMC was hanged this bit will be turn off.

5.13 GRAPH1SYS CONFIG Register

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CONFG_BASE + 300h	Clock Gating Control Status Register	GRAPH1SYS_CG_CON
CONFG_BASE + 320h	Clock Gating Set Register	GRAPH1SYS_CG_SET
CONFG_BASE + 340h	Clock Gating Clear Register	GRAPH1SYS_CG_CLR
CONFG_BASE + 400h	LCD IO Selection	GRAPH1SYS_LCD_IO_SEL
CONFG_BASE + 600h	Memory Delsel Control Regsiter 0	GRAPH1SYS_DELSEL0
CONFG_BASE + 604h	Memory Delsel Control Regsiter 1	GRAPH1SYS_DELSEL1
CONFG_BASE + 608h	Memory Delsel Control Regsiter 2	GRAPH1SYS_DELSEL2
CONFG_BASE + 60Ch	Memory Delsel Control Register 3	GRAPH1SYS_DELSEL3

Table 101 ***APB Bridge Register Map***

5.13.1 Register Definitions

CONFG_BASE + 300h Clock Gating Control Status Register GRAPH1SYS_CG_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							RO	RO	RO	RO		RO	RO		RO	RO
Reset							1	1	1	1		1	1		1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGD MA0	BLS	GCMQ	G2D	GMC1
Type		RO	RO	RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO
Reset		1	1	1	1	1	1	1		1	1	1	1	1	1	0

GRAPH1 sub-system clock gating control status register (read only), value 1 represents clock gating.

GMC1 Status of the GMC1 Clock Gating.
G2D Status of the G2D Clock Gating.
GCMQ Status of the GCMQ Clock Gating.
BLS Status of the BLS Clock Gating.
IMGDMA0 Status of the IMGDMA0 Clock Gating.
PNG Status of the PNG Clock Gating.
DSI Status of the DSI Clock Gating.
TVE Status of the TVE Clock Gating.
TVC Status of the TVC Clock Gating.
ISP Status of the ISP Clock Gating.
IPP Status of the IPP Clock Gating.
PRZ Status of the PRZ Clock Gating.
CRZ Status of the CRZ Clock Gating.
DRZ Status of the DRZ Clock Gating.
WT Status of the WT Clock Gating.
AFE Status of the AFE Clock Gating.
SPI Status of the SPI Clock Gating.
ASM Status of the ASM Clock Gating.
RESZ_LB Status of the RESZ_LB Clock Gating.
LCD Status of the LCD Clock Gating.
DPI Status of the DPI Clock Gating.
G1FAKE Status of the G1FAKE Clock Gating.

CONFG_BASE + 320h Clock Gating Set Register GRAPH1SYS_CG_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							WO	WO	WO	WO		WO	WO		WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGD MA0	BLS	GCMQ	G2D	GMC1
Type		WO	WO	WO	WO	WO	WO	WO		WO	WO	WO	WO	WO	WO	WO

GRAPH1 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding "SET" register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_COND registers.

Eg.

If PDN_COND = 16'h0F0F,

Writing PDN_COND = 16'F0F0 will result in PDN_COND = 16'hFFFF.

 © 2009 MediaTek Inc.

GMC1 Set GMC1 Clock Gating.
G2D Set G2D Clock Gating.
GCMQ Set GCMQ Clock Gating.
BLS Set BLS Clock Gating.
IMGDMA0 Set IMGDMA0 Clock Gating.
PNG Set PNG Clock Gating.
DSI Set DSI Clock Gating.
TVE Set TVE Clock Gating.
TVC Set TVC Clock Gating.
ISP Set ISP Clock Gating.
IPP Set IPP Clock Gating.
PRZ Set PRZ Clock Gating.
CRZ Set CRZ Clock Gating.
DRZ Set DRZ Clock Gating.
WT Set WT Clock Gating.
AFE Set AFE Clock Gating.
SPI Set SPI Clock Gating.
ASM Set ASM Clock Gating.
RESZ_LB Set RESZ_LB Clock Gating.
LCD Set LCD Clock Gating.
DPI Set DPI Clock Gating.
G1FAKE Set G1FAKE Clock Gating.

CONFG_BASE + 340h Clock Gating Clear Register GRAPH1SYS_CG_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							G1FAKE	DPI	LCD	RESZ_LB		ASM	SPI		AFE	WT
Type							WO	WO	WO	WO		WO	WO		WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DRZ	CRZ	PRZ	IPP	ISP	TVC	TVE		DSI	PNG	IMGDMA0	BLS	GCMQ	G2D	GMC1
Type		WO	WO	WO	WO	WO	WO	WO		WO	WO	WO	WO	WO	WO	WO

GRAPH1 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding PDN_COND registers.
Eg.
If PDN_COND = 16’hFFFF,
Writing PDN_COND = 16’F0F0 will result in PDN_COND = 16’h0F0F.

GMC1 Clear GMC1 Clock Gating.
G2D Clear G2D Clock Gating.
GCMQ Clear GCMQ Clock Gating.
BLS Clear BLS Clock Gating.

IMGDMA0	Clear IMGDMA0 Clock Gating.
PNG	Clear PNG Clock Gating.
DSI	Clear DSI Clock Gating.
TVE	Clear TVE Clock Gating.
TVC	Clear TVC Clock Gating.
ISP	Clear ISP Clock Gating.
IPP	Clear IPP Clock Gating.
PRZ	Clear PRZ Clock Gating.
CRZ	Clear CRZ Clock Gating.
DRZ	Clear DRZ Clock Gating.
WT	Clear WT Clock Gating.
AFE	Clear AFE Clock Gating.
SPI	Clear SPI Clock Gating.
ASM	Clear ASM Clock Gating.
RESZ_LB	Clear RESZ_LB Clock Gating.
LCD	Clear LCD Clock Gating.
DPI	Clear DPI Clock Gating.
G1FAKE	Clear G1FAKE Clock Gating.

CONFG_BASE + 400h LCD IO Selection GRAPH1SYS_LCD_IO_SEL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															LCD_IO_SEL	
Type															R/W	
Reset															0	0

LCD_IO_SEL Pin NLD[25:0] are shared between CPU interface and RGB interface. Use LCD_IO_SEL to select the sharing mode as Table 2.

lcd_io_sel	NLD bits	0~7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2'B00	CPU IF only	D0~7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23		
2'B01	8-bit CPU+18-bit RGB	D0~7	B0	B1	B2	B3	B4	B5	G0	G1	G2	G3	G4	G5	R0	R1	R2	R3	R4	R5
2'B10	9-bit CPU+16-bit RGB	D0~7	B0	B1	B2	B3	B4	G0	G1	G2	G3	G4	G5	R0	R1	R2	R3	R4	D8	
2'B11	18-bit CPU+8-bit RGB	D0~7	D8	D9	D10	D11	D12	D13	D14	D15	B0	B1	B2	B3	B4	B5	B6	B7	D16	D17

(pink): CPU IF

(green): RGB IF

Table 102 ***LCD_IO_SEL mapping table***

CONFG_BASE + 600h Memory delsel control register 0 GRAPH1SYS_DELSEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ISP_NR1MEM_DELSEL[5:0]						ISP_SHADMEM_DELSEL[1:0]		RESZ_LB_DELSEL[1:0]		IMGDMA_DELSEL[1:0]		LCD_CACHE_DELSEL[1:0]		LCD_WBUF_DELSEL[1:0]	

Type	RW						RW		RW		RW		RW		RW	
Reset	6'h22						2'h2		2'h3		2'h2		2'h3		2'h0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LCD_PAL_DELSEL[1:0]		LCD_DSIFIFO_DELSEL[1:0]		TVC_RAM_DELSEL[1:0]		WAVE_DELSEL[1:0]		ASM_DELSEL[1:0]		AFE_DELSEL[1:0]		DELSEL_FIFO_MEM[1:0]		DELSEL_CURSOR_MEM[1:0]	
Type	RW		RW		RW		RW		RW		RW		RW		RW	
Reset	2'h3		2'h1		2'h3		2'h2		2'h2		2'h2		2'h1		2'h2	

CONFG_BASE + 604h — Memory delsel control register 1 — GRAPH1SYS_DELSEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ISP_DMMEM0_DELSEL[5:0]						ISP_DMMEM1_DELSEL[15:6]									
Type	RW						RW									
Reset	6'h22						10'h88									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISP_DMMEM1_DELSEL[5:0]						ISP_NR1MEM_DELSEL[15:6]									
Type	RW						RW									
Reset	6'h22						10'h88									

CONFG_BASE + 608h — Memory delsel control register 2 — GRAPH1SYS_DELSEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ISP_AWBHISGMEM_DELSEL[1:0]		ISP_AWBHISRMEM_DELEL[1:0]		ISP_AWBMEM_DELSEL[1:0]		ISP_COLORMEM_DELSEL[7:0]								ISP_DMMEM0_DELSEL[23:22]	
Type	RW		RW		RW		RW								RW	
Reset	2'h2		2'h2		2'h2		8'h3A								2'h0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISP_DMMEM0_DELSEL[21:6]															
Type	RW															
Reset	16'h88															

CONFG_BASE + 60Ch — Memory delsel control register 3 — GRAPH1SYS_DELSEL3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSI_DELSEL[1:0]		GM1_DELSEL[7:0]								ISP_3AMEM_DELSEL[1:0]		ISP_AEHISMEM_DELSEL[1:0]		ISP_AWBHISBMEM_DELSEL[1:0]	
Type	RW		RW								RW		RW		RW	
Reset	2'h2		8'hFF								2'h2		2'h2		2'h2	

5.13.2 GRAPH1SYS Application Note

5.13.2.1 Power up sequence for LCD, DPI, and DSI (clocked at AHB domain)

For registers clocked at AHB clock domain (LCD, DPI, and DSI), please follow below power up sequence to ensure availability of register read.

1. Clear power down bits of LCD, DPI, or DSI
2. Add **40** NOPs
3. Register read (standard processes)

5.14 GRAPH2SYS CONFIG Register

REGISTER ADDRESS	REGISTER NAME	SYNONYM
CONFG_BASE + 000h	Clock Gating Control Status Register	GRAPH2SYS_CG_CON
CONFG_BASE + 004h	Clock Gating Set Register	GRAPH2SYS_CG_SET
CONFG_BASE + 008h	Clock Gating Clear Register	GRAPH2SYS_CG_CLR
CONFG_BASE + 010h	Memory Delsel Control Regsiter 0	GRAPH2SYS_DELSEL0
CONFG_BASE + 014h	Memory Delsel Control Regsiter 1	GRAPH2SYS_DELSEL1
CONFG_BASE + 018h	Memory Delsel Control Regsiter 2	GRAPH2SYS_DELSEL2

Table 103 ***APB Bridge Register Map***

5.14.1 Register Definitions

CONFG_BASE + 000h **Clock Gating Control Status Register** **GRAPH2SYS_CG_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset								1	1	1	1	1	1	1	1	0

GRAPH2 sub-system clock gating control status register (read only), value 1 represents clock gating.

GMC2 Status of the GMC2 Clock Gating.

IMAGE_DMA_1 Status of the IMAGE_DMA_1 Clock Gating.

PRZ Status of the PRZ Clock Gating.

 © 2009 MediaTek Inc.

M3D Status of the M3D Clock Gating.
H264 Status of the H264 Clock Gating.
DCT Status of the DCT Clock Gating.
JPEG Status of the JPEG Clock Gating.
MP4 Status of the MP4 Clock Gating.
MP4_DEBLK Status of the MP4_DEBLK Clock Gating.

CONFG_BASE + 004h Clock Gating Set Register GRAPH2SYS_CG_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2
Type								W/O	WO	WO	WO	WO	WO	WO	WO	WO

GRAPH2 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “SET” register will perform a bit-wise **OR** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.
Eg.
If CG_CON = 16’h0F0F,
Writing CG_SET = 16’F0F0 will result in CG_CON = 16’hFFFF.

GCU Set the GCU Controller Power Down.
GMC2 Set the GMC2 Clock Gating.
IMAGE_DMA_1 Set the IMAGE_DMA_1 Clock Gating.
PRZ Set the PRZ Clock Gating.
M3D Set the M3D Clock Gating.
H264 Set the H264 Clock Gating.
DCT Set the DCT Clock Gating.
JPEG Set the JPEG Clock Gating.
MP4 Set the MP4 Clock Gating.
MP4_DEBLK Set the MP4_DEBLK Clock Gating.

CONFG_BASE + 008h Clock Gating Clear Register GRAPH2SYS_CG_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								MP4_DEBLK	MP4	JPEG	DCT	H264	M3D	PRZ	IMAGE_DMA_1	GMC2
Type								WO	WO	WO	WO	WO	WO	WO	WO	WO

GRAPH2 sub-system clock gating set register, value 1 represents clock gating. For all registers addresses listed above, writing to the corresponding “Clear” register will perform a bit-wise **AND-NOT** function between the 32bit written value and the 32bit register value already existing in the corresponding CG_CON registers.
Eg.
If CG_CON = 16’hFFFF,
Writing CG_CLR = 16’F0F0 will result in CG_CON = 16’h0F0F.

GCU Clear the GCU Controller Power Down.
GMC2 Clear the GMC2 Clock Gating.
IMAGE_DMA_ Clear the IMAGE_DMA_1 Clock Gating.
PRZ Clear the PRZ Clock Gating.
M3D Clear the M3D Clock Gating.
H264 Clear the H264 Clock Gating.
DCT Clear the DCT Clock Gating.
JPEG Clear the JPEG Clock Gating.
MP4 Clear the MP4 Clock Gating.
MP4_DEBLK Clear the MP4_DEBLK Clock Gating.

CONFG_BASE + 010h Memory delsel control register 0 GRAPH2SYS_DELSEL0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D[31:16]**															
Type	RW															
Reset	16’H002A															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D[15:0]**															
Type	RW															
Reset	16’HAAA9															

CONFG_BASE + 014h Memory delsel control register 1 GRAPH2SYS_DELSEL1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D[63:48]**															
Type	RW															
Reset	16’H5500															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D[47:32]**															
Type	RW															
Reset	16’H0000															

CONFG_BASE + 018h Memory delsel control register 2 GRAPH2SYS_DELSEL2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**GMC2[11:10]**	
Type															RW	
Reset															2’H3	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GMC2[9:0]	M3D[69:64]
Type	RW	RW
Reset	10'H3FF	6'H2D

5.15 H.264 Decoder

5.15.1 General Description

The H.264 videos, is getting more and more popular in today's application for its high coding efficiency. In order to play the H.264 videos in high quality, the hardware H.264 decoder is developed. For the mobile application, we implemented the whole scope of the baseline profile in H.264 standard including I-slice, P-slice, CAVLC, Slice-Group, ASO, Redundant Slices, and so on. We also made our H.264 decoder standard compliant.

The scope of H.264 hardware decoder is from Slice Data Layer and below, which are the main part of the H.264 standard. Thus, before trigger this H.264 decoder to decode, software have to parse and processing the part of slice header and above. For a multiple-slice with slice group video, the slice header and above contain information of video size, reference frames ordering, slice length of each slice, and slice index of each macroblock (slice map), and so on. After parsing the information in slice-layer and above, software has to fill the following information to hardware for decoding this frame.

(1) The memory address of "slice_configuration" : **H264_DEC_SLICE_CONF**

(2) For each slice, fill a 32-bytes data into memory from the start address of "slice_configuration". 32-bytes by 32-bytes continuous in memory address.

(3) Each "slice_configuration" contains the following information (Fig. 2):
(a) slice starting address
(b) slice starting bitcnt
(c) slice length
(d) reference frame index list for this slice
(e) some parameters like filteroffset, qp, and etc in slice header.

(4) Decoding the slice map and fill the slice map into memory starting from "**H264_DEC_SLICE_MAP_ADDR**"

(5) Fill the reference frame starting address 4-bytes by 4-bytes starting from "**H264_REF_FRAME_ADDR**"

(6) Allocate working memory for hardware:
(a) "**H264_MC_LINE_BUF_ADDR**": 16-bytes aligned, size="**WIDTH_in_MB*16 * 48 *1.5 Bytes**", internal memory, this memory is only for prefetch enabled case.
(b) "**H264_MC_MV_BUFFER_ADDR**": 16-bytes aligned, size=3328 bytes, internal memory recommended.
(c) "**H264_DEC_REC_ADDR**": 16-bytes aligned, size=" **WIDTH_in_MB *16* HEIGHT_in_MB*16 *1.5 bytes**", external memory recommended.
(d) "**H264_DEC_DEB_BUF_ADDR**": 16-bytes aligned, size=" **(WIDTH_in_MB *8 +8)*16 bytes**", internal memory recommended
(e) "**H264_DEC_DEB_DAT_BUF0_ADDR**": 512-bytes aligned, size=384 bytes, internal memory

recommended.
(f) “**H264_DEC_DEB_DAT_BUF1_ADDR**”: 512-bytes aligned, size=384 bytes, internal memory recommended.
(g) “**H264_CAVLC_BASE_ADDR**”: 32-bytes aligned, size=520 bytes, internal memory recommended.

After programming the above information, the operation of H.264 hardware decoder will be described as the follows:

(1) For each macroblock, first read the slice index of this macroblock from slicemap.

(2) Reading the associated slice configuration for this macroblock according to slice index.

(3) If this macroblock is I-block, then do the scaling, IDCT.

(4) If this macroblock is P-block, besides residual value decoded same to I-block, motion compensation is applied. The motion compensation procedure are:
(a) read reference frame index from first 8 bytes in the associated slice configuration according to the decoded ref_idx.
(b) read the reference frame start address from the “**H264_REF_FRAME_ADDR**” according to the reference frame index read at (a).
(c) doing motion compensation from the frame of base address read at (b)

(5) Performing de-blocking filter operation.

To speed up the memory access of motion compensation, the H.264 decoder supports memory prefetch scheme. A hardware module moves pixel data from external memory to internal memory continuously in decoding pipeline, such that the motion compensation module can do memory fetching from internal memory if the motion vector lies in the prefetch memory. To set “mc_pfhen”=1 in “**H264_MC_LINE_BUF_OFFSET**” to enable this function.

After H.264 decoder is being triggered, the decoder will send interrupt in the following case:

(1) Decoding complete successfully : “DEC_DONE” flag will be set to 1

(2) DMA reached DMA_LIMIT and paused : “DMA_PAUSE” flag will be set to 1

(3) VLD decoding error : “VLD_ERROR” flag will be set to 1

(4) IDCT overflow error : “OVERFLOW” flag will be set to 1

(5) Invalid MB type decoded : “MB_TYPE_ERR” flag will be set to 1

(6) Macroblock location just hit the MB_POS set : “MB_POS_IRQ” flag will be set to 1

These flags can be observed in the register “**H264_DEC_IRQ_STS**”.

After receiving these interrupt from hardware, software have to acknowledge these interrupt by setting “H264_DEC_IRQ_ACK” to clear the interrupt status, and then perform associated actions.

5.15.2 Registers Definitions

Register Address	Register Function	Acronym
H264+0000h	H264 Decoder Command Register	**H264_DEC_COMD**

H264+0004h	H264 Decoder DMA Command Register	**H264_DEC_DMA_COMD**
H264+0008h	H264 Decoder Slice Group Mapping Register	**H264_DEC_SLICE_MAP_ADDR**
H264+000Ch	H264 Decoder Picture Level Configuration Register	**H264_DEC_PIC_CONF**
H264+0010h	H264 Decoder Slice Level Configuration Register	**H264_DEC_SLICE_CONF**
H264+0020h	H264 Decoder DMA Limit Register	**H264_DEC_DMA_LIMIT**
H264+0030h	H264 Reference Frame Address Register	**H264_REF_FRAME_ADDR**
H264+0034h	H264 MC Line Buffer Address Register	**H264_MC_LINE_BUF_ADDR**
H264+0038h	H264 MC Line Buffer Offset Register	**H264_MC_LINE_BUF_OFFSET**
H264+003Ch	H264 MC Line Buffer Size Register	**H264_MC_LINE_BUF_SIZE**
H264+0040h	H264 MC Motion Vector Buffer Address Register	**H264_MC_MV_BUFFER_ADDR**
H264+0044h	H264 Reconstruction Address Register	**H264_DEC_REC_ADDR**
H264+0048h	H264 Deblk Buffer Address Register	**H264_DEC_DEB_BUF_ADDR**
H264+004Ch	H264 Reconstruction Luma Size Register	**H264_DEC_REC_Y_SIZE**
H264+0050h	H264 Deblk Data Buffer0 Address Register	**H264_DEC_DEB_DAT_BUF0_ADDR**
H264+0054h	H264 Deblk Data Buffer1 Address Register	**H264_DEC_DEB_DAT_BUF1_ADDR**
H264+0060h	H264 CAVLC Base Address Register	**H264_CAVLC_BASE_ADDR**
H264+0070h	H264 Decoder Interrupt Status Register	**H264_DEC_IRQ_STS**
H264+0074h	H264 Decoder Interrupt Mask Register	**H264_DEC_IRQ_MASK**
H264+0078h	H264 Decoder Interrupt Acknowledge Register	**H264_DEC_IRQ_ACK**
H264+007Ch	H264 Decoder MB_ IRQ_POS Setting	**H264_DEC_MB_IRQ_POS**
H264+0100h	H264 Decoder DMA Status Register	**H264_DEC_DMA_STS**
H264+0104h	H264 Decoder Debug INFO0 Register	**H264_DEC_DEBUG_INFO0**
H264+0108h	H264 Decoder Debug INFO1 Register	**H264_DEC_DEBUG_INFO1**

H264+010Ch	H264 Decoder Debug INFO2 Register	**H264_DEC_DEBUG_INFO2**
H264+0110h	H264 Decoder Debug INFO3 Register	**H264_DEC_DEBUG_INFO3**
H264+0114h	H264 Decoder Debug INFO4 Register	**H264_DEC_DEBUG_INFO4**
H264+0118h	H264 Decoder Debug INFO5 Register	**H264_DEC_DEBUG_INFO5**
H264+011Ch	H264 Decoder Debug INFO6 Register	**H264_DEC_DEBUG_INFO6**
H264+0120h	H264 Decoder Debug INFO7 Register	**H264_DEC_DEBUG_INFO7**
H264+0124h	H264 Decoder Debug INFO8 Register	**H264_DEC_DEBUG_INFO8**
H264+0128h	H264 Decoder Debug INFO9 Register	**H264_DEC_DEBUG_INFO9**
H264+012Ch	H264 Decoder Debug INFO10 Register	**H264_DEC_DEBUG_INFO10**
H264+0130h	H264 Decoder Debug INFO11 Register	**H264_DEC_DEBUG_INFO11**

Table 104 ***H.264 Decoder Registers***

5.15.2.1 Main Configuration & Commands

H264+0000h H264 Decoder Command Register H264_DEC_COMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**DEC_START**				**DEC_RST**
Type												WO				WO

DEC_RST Software reset control for H.264 video DECODER. The device driver software must always set this bit to 1 before starting decoding procedure.

DEC_START Start the decoding operation. Set this bit will trigger H.264 decoder.

H264+0004h H264 Decoder DMA Command Register H264_DEC_DMA_COMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**RESUME**	**STOP**
Type															WO	WO

STOP Stop the DMA. Stop DMA activities through SW rather than HW state machine

RESUME Resume the DMA access. DMA state machine will go to a pending state if the maximum allowed write count to target memory is reached and then an interrupt has occured. After re-allocating the target address, SW writes RESUME to unfreeze the encoding process.

H264+0008h H264 Decoder Slice Group Mapping Register H264_DEC_SLICE_MAP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SLICE_MAP_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SLICE_MAP_ADDR[15:13]															
Type	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

SLICE_MAP_ADDR The address of memory buffer storing slice group mapping; note that the lower **13 bits** of this address must be all zeros. The buffer size to store slice group mapping is **(1024/16) * HEIGHT_in_MB * 2 Bytes**. (3840Bytes for VGA)

H264+000Ch H264 Decoder Picture Level Configuration Register H264_DEC_PIC_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						MF_CACHE	DMA_CACHE	LAST_REF_IDX[3:0]				REF_CUR_FRAME	FRAME_TYPE	ERR_STALL_EN	INTR_FLAG	IRQ_EN
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PIC_HEIGHT								PIC_WIDTH					
Type			R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0			0	0	0	0	0	0

PIC_WIDTH Picture width in the unit of macroblocks is ***N***, please program ***(N-1)***.

PIC_HEIGHT Picture height in the unit of macroblocks is ***M***, please program ***(M-1)***.

Note : the maximum supported picture width and height is 992 pixels, and the maximum number of macroblock supported is 2047.

IRQ_EN Set to 1 is to enable Interrupt.

INTRA_FLAG Constrained Intra Perdition Flag extract from header.

ERR_STALL_EN Set to 1 to stall H.264 Decoder when VLD_ERROR or IDCT_OVERFLOW =1

FRAME_TYPE Set to 1 if any slice in this frame is P slice

REF_CUR_FRM 1 bit to indicate if the current frame is used for reference.

LAST_REF_IDX indicate the last reference frame in the current reference list.

DM A_CACHE This bit should be set to 0.

MF_CACHE This bit should be set to 1.

H264+0010h H264 Decoder Slice Level Configuration Register H264_DEC_SLICE_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SLICE_CONF_ADDR[31:16]															

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SLICE_CONF_ADDR[15:5]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

SLICE_CONF_ADDR Starting address in memory storing slice level configurations for all slices and must be **32-bytes** aligned. This register contains the slice-level configuration information for all slices. Each slice will consume 32 bytes to store its configuration. So the first 32 bytes is the configuration field for slice with slice index equal to 0; the next 32 bytes is for slice with slice index equal to 1, and so on. In each configuration field, the following information is put in the order of ascending address:

REF_FRAME_INDEX_LIST 8 bytes for 16 reference frames and LSB-4bit of 1st byte to indicate the idex0 of L0. Each reference frame index is 4 bits. The reference frame index at position 0 in this field is for the first reference frame in reference frame list L0; the reference frame index at position 1 in this field is for the second reference frame in reference frame list L0, and so on.

SLICE_LENGTH Slice bit stream length in bytes; used for error handling. 4 bytes in length.

SLICE_START_ADDR The real starting **4-byte aligned** address of slice bit stream in memory. 4 bytes in length.

SLICE_START_BIT_CNT With SLICE_START_ADDR defining the start byte address of slice bit stream, SLICE_START_BIT_CNT specifies which bit in this address hardware should begin to parse. 6-bit in length.

FILTEROFFSETA Signed number. 5-bit in length. In standard, FilterOffsetA = slice_alpha_c0_offset_div2 << 1, where slice_alpha_c0_offset_div2 is directly parsed from slice header.

FILTEROFFSETB Signed number. 5-bit in length. In standard, FilterOffsetB = slice_beta_offset_div2 << 1, where slice_beta_offset_div2 is directly parsed from slice header.

INIT_SLICE_QP 1 byte in length. INIT_SLICE_QP is the initial slice quantization parameter used for scaling & transform. INIT_SLICE_QP = 26 + pic_init_qp_minus26 + slice_qp_delta. pic_init_qp_minus26 is directly parsed from pic_parameter_set_rbsp and slice_qp_delta is directly parsed from slice header.

REC_LIST_IDX 4 bits in length. Indicate the reference list0 index in the current slice of the last reconstruction frame.

VALID 1bits to indicate if the rec_list_idx is valid

DISABLE_DEBLOCKING_FILTER_IDC 2 bit in length. Disable_deblocking_filter_idc can be directly parsed from slice header.

SLICE_TYPE 1 bit in length. SLICE_TYPE is directly parsed from slice header.

OLD_SLICE 1 bit in length. OLD_SLICE should be set to zero before starting one frame.

NUM_REF_IDX_L0_ACTIVE_MINUS1 4-bit in length to indicate the number of the active reference frame.

CHROMA_QP_INDEX_OFFSET 5-bit in length to indicate qp-offset of chroma.

With all fields shown above, around 21 of 32 bytes are used. The rest will be used as hardware intermediate data buffer between slice switching.

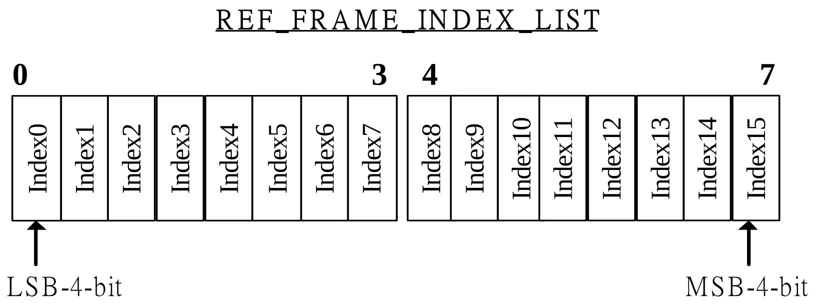


Fig. 1. Refernce frame index list

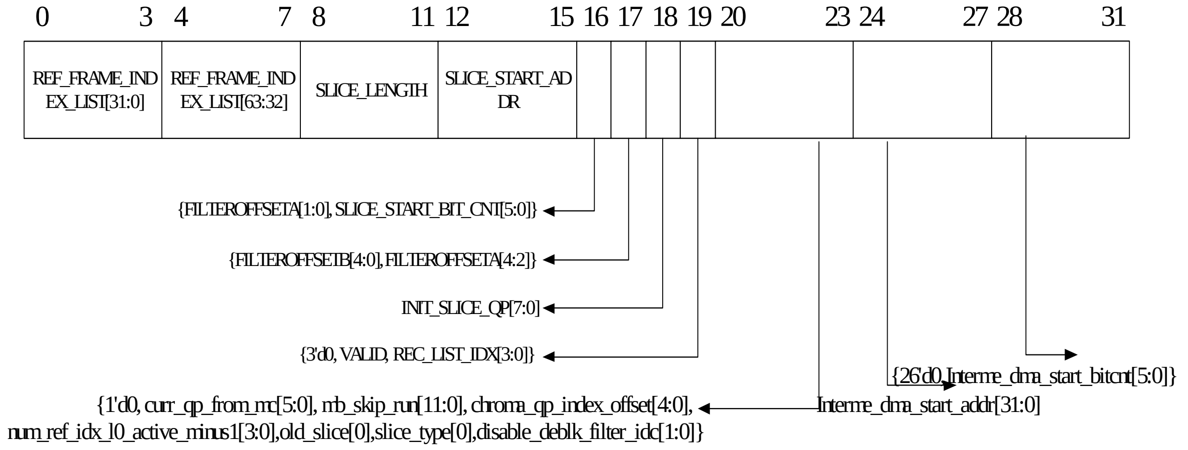


Fig. 2.

structure of slice configuration

5.15.2.2 DMA

H264+0020h H264 Decoder DMA Limit Register — H264_DEC_DMA_LIMIT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMA_LIMIT[16:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register is used to describe the buffer size of each VLC DMA buffer. After DMA consume 4 bytes bit-streams, DMA_LIMIT will decrease one. Whenever the limit is reached (decrease as resuming the bitstream) and DMA_PAUSE interrupt mask is off, and DMA_PAUSE interrupt will be generated.

5.15.2.3 MC

H264+0030h H264 Reference Frame Address Register – All Slice Groups H264_REF_FRAME_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**REF_PIC_ADDR[31:16]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**REF_PIC_ADDR[15:6]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO

This register specifies an address where the actual locations in SRAM of 16 reference frames are stored. The base address must be **64-bytes aligned**. The real locations of 16 reference frames are implied as follow:
Address of reference frame with reference frame index equal to 0x0 is REF_PIC_ADDR[31:0].
Address of reference frame with reference frame index equal to 0x1 is (REF_PIC_ADDR+4)[31:0].
Address of reference frame with reference frame index equal to 0x2 is (REF_PIC_ADDR+8)[31:0].
Address of reference frame with reference frame index equal to 0x3 is (REF_PIC_ADDR+12)[31:0].
Address of reference frame with reference frame index equal to 0x4 is (REF_PIC_ADDR+16)[31:0].
Address of reference frame with reference frame index equal to 0x5 is (REF_PIC_ADDR+20)[31:0].
Address of reference frame with reference frame index equal to 0x6 is (REF_PIC_ADDR+24)[31:0].
Address of reference frame with reference frame index equal to 0x7 is (REF_PIC_ADDR+28)[31:0].
Address of reference frame with reference frame index equal to 0x8 is (REF_PIC_ADDR+32)[31:0].
Address of reference frame with reference frame index equal to 0x9 is (REF_PIC_ADDR+36)[31:0].
Address of reference frame with reference frame index equal to 0xa is (REF_PIC_ADDR+40)[31:0].
Address of reference frame with reference frame index equal to 0xb is (REF_PIC_ADDR+44)[31:0].
Address of reference frame with reference frame index equal to 0xc is (REF_PIC_ADDR+48)[31:0].
Address of reference frame with reference frame index equal to 0xd is (REF_PIC_ADDR+52)[31:0].
Address of reference frame with reference frame index equal to 0xe is (REF_PIC_ADDR+56)[31:0].
Address of reference frame with reference frame index equal to 0xf is (REF_PIC_ADDR+60)[31:0].

REF_PIC_ADDR Base address storing the location of 16 reference frames, and the base address of every reference frame must be **32-bit** aligned.

H264+0034h H264 MC Line Buffer Address Register H264_MC_LINE_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MC_LINE_BUF_ADDR[31:16]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MC_LINE_BUF_ADDR[15:4]**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

MC_LINE_BUF_ADDR Base address for mc pre-fetch line buffer and **16-bytes aligned**. The pre-fetch buffer size for mc is **WIDTH_in_MB*16 * 48 *1.5 Bytes**. (46080Bytes for VGA)

H264+0038h H264 MC Line Buffer Offset Register H264_MC_LINE_BUF_OFFSET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																MC_PFHEN
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										UVBUF_OFFSET				YBUF_OFFSET		
Type										R/W	R/W	R/W		R/W	R/W	R/W

MC_PFH_EN Enable MC pre-fetch pixel data mode if MC_PFH_EN = 1'b1.
Note : This prefetch mode can only be enabled under the case that WIDTH_IN_MB>=3 && HEIGHT_IN_MB>=4

UVBUF_OFFSET Number of MC pre-fetch Line buffer for chroma. This number must be 2.

YBUF_OFFSET Number of MC pre-fetch Line buffer for luma. This number must be 4.

H264+003Ch H264 MC Line Buffer Size Register H264_MC_LINE_BUF_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MC_LINE_UVBUF_SIZE[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MC_LINE_YBUF_SIZE[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC_LINE_UVBUF_SIZE Chroma Buffer byte size for mc pre-fetch line buffer. The pre-fetch uvbuffer size for mc is **WIDTH_in_MB* (8+8*MC_LINE_UVBUF_OFFSET) bytes** and must be **multiples of 4 bytes**. (30720 Bytes for VGA as MC_LINE_BUF_OFFSET =2)

MC_LINE_YBUF_SIZE Luma Buffer byte size for mc pre-fetch line buffer. The pre-fetch ybuffer size for mc is **WIDTH_in_MB* (16+8*MC_LINE_YBUF_OFFSET) bytes** and must be **multiples of 4 bytes**. (30720 Bytes for VGA as MC_LINE_BUF_OFFSET =4)

H264+0040h H264 MC Motion Vector Buffer Address Register H264_MC_MV_BUFFER_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MC_MV_BUF_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MC_MV _BUF_ADDR[15:4]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

MC_MV_BUF_ADDR Base address for mc motion vector buffer and must be **16-byte aligned**. Total memory is fixed to **3328 bytes.**

5.15.2.4 Deblocking Filter

H264+0044h H264 Reconstruction Address Register H264_DEC_REC_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	REC_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REC_ADDR [15:4]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

REC_ADDR Reconstruction address. The address must be **16-bytes aligned**. The memory size needs **WIDTH_in_MB *16* HEIGHT_in_MB*16 *1.5 bytes** (460800 bytes for VGA)

H264+0048h H264 Deblk Buffer Address Register H264_DEC_DEB_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEB_BUF_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEB_BUF_ADDR[15:13]															
Type	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

DEB_BUF_ADDR Deblocking filter buffer working buffer. The address must be **16-bytes aligned.** The memory size needs **(WIDTH_in_MB *8 +8)*16 bytes** (5248 bytes for VGA)

H264+004Ch H264 Reconstruction Luma Size Register H264_DEC_REC_Y_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													REC_Y_SIZE[19:16]			
Type													R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REC_Y_SIZE[15:4]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

REC_Y_SIZE Size of Reconstruction memory for luma and the supported size must be multiples of **16-bytes** . The size is **(WIDTH_in_MB *16* HEIGH _in_MB*16) bytes** (307200 bytes for VGA)

H264+0050h H264 Deblk Data Buffer0 Address Register H264_DEC_DEB_DAT_BUF0_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEBLK_DAT_BUF0_ADDR[16:15]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEBLK_DAT_BUF0_ADDR[15:4]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

DEBLK_DAT_BUF0_ADDR Deblocking filter/MC hand-shaking buffer0 address and must be **16-bytes aligned**. The size is **384 bytes**. If **DEBLK_DATA_BUF0_ADDR** was assigned different internal memory bank address from **DEBLK_DATA_BUF1_ADDR**, decoding performance could be better.

NOTE: If video sequence supports IPCM, the address must be 512-bytes aligned.

H264+0054h H264 Deblk Data Buffer1 Address Register H264_DEC_DEB_DAT_BUF1_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEBLK_DATA_BUF0_ADDR[16:15]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEBLK_DATA_BUF0_ADDR[15:4]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

DEBLK_DAT_BUF1_ADDR Deblocking filter/MC hand-shaking buffer0 address and must be **16-bytes aligned**. The size is **384 bytes**. If **DEBLK_DATA_BUF1_ADDR** was assigned different internal memory bank address from **DEBLK_DATA_BUF0_ADDR**, decoding performance could be better.

NOTE: If video sequence supports IPCM, the address must be 512-bytes aligned.

5.15.2.5 CAVLC

H264+0060h H264 CAVLC Base Address Register H264_CAVLC_BASE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CAVLC_BASE_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAVLC_BASE_ADDR[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CAVLC_BASE_ADDR CAVLC working buffer base address and must be **32-bit aligned**. Total size is **520 bytes.**

5.15.2.6 Interrupts

H264+0070h H264 Decoder Interrupt Status Register H264_DEC_IRQ_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											MB_POS_IRQ	MB_TYPE_ER	OVERFLOW	VLD-ERROR	DMA_PAUSE	DEC_DONE
Type											R/W	R/W	R/W	R/W	R/W	R/W

DEC_DONE Decode complete interrupt. Goes high when decoder operation is done.

DMA_PAUSE DMA pause interrupt. Goes high when DMA access length reaches DMA limit. DMA access will resume when software program a new DMA address into hardware..

VLD_ERROR VLD error interrupt. Goes high when CAVLC decoder returns error as processing bit stream. This interrupt implies an error in bitstream.

OVERFLOW Overflow interrupt. Goes high when decoder length counter reaches slice length and decoding operation is still not over. This interrupt implies an error in bitstream.

MB_TYPE_ERR MB type error interrupt. Goes high when decoded mb_type is an invalid type.
MB_POS_IRQ MB position interrupt. Goes high when current decoding macroblock matches the setting of H264_DEC_MB_IRQ_POS.

H264+0074h H264 Decoder Interrupt Mask Register H264_DEC_IRQ_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											MB_POS_IRQ	MB_TYPE_ER	OVERFLOW	VLD_ERROR	DMA_PAUSE	DEC_DONE
Type											R/W	R/W	R/W	R/W	R/W	R/W
Reset											0	0	0	0	0	0

DEC_DONE Mask of decoder complete interrupt and interrupt status.
DMA_PAUSE Mask of DMA pause interrupt and interrupt status.
VLD_ERROR Mask of VLD error interrupt and interrupt status.
OVERFLOW Mask of overflow interrupt and interrupt status from IDCT.
MB_TYPE_ERR Mask of overflow interrupt and interrupt status of MB_TYPE_ERR.
MB_POS_IRQ Mask of overflow interrupt and interrupt status of MB_POS_IRQ.

H264+0078h H264 Decoder Interrupt Acknowledge Register H264_DEC_IRQ_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											MB_POS_IRQ	MB_TYPE_ER	OVERFLOW	VLD_ERROR	DMA_PAUSE	DEC_DONE
Type											R/W	R/W	R/W	R/W	R/W	R/W

DEC_DONE Decoder complete interrupt acknowledge.
DMA_PAUSE DMA pause interrupt acknowledge.
VLD_ERROR VLD error interrupt acknowledge.
OVERFLOW Overflow interrupt acknowledge from IDCT.
MB_TYPE_ERR Overflow interrupt acknowledge of MB_TYPE_ERR.
MB_POS_IRQ Overflow interrupt acknowledge of MB_POS_IRQ.

H264+007Ch H264 Decoder MB_ IRQ_POS Setting H264_DEC_MB_IRQ_POS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			MB_IRQ_POS_Y								MB_IRQ_POS_X					

Type			R/W								R/W					

MB_IRQ_POS_X, MB_IRQ_POS_Y MB_POS_IRQ interrupt will generate when current decoding Macroblocks is at the location of (MB_IRQ_POS_Y,MB_IRQ_POS_X).

5.15.2.7 Debug Information

H264+0100h H264 Decoder DMA Status Register H264_DEC_DMA_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				GADDR[15:3]												
Type				RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GADDR[2:0]					WAIT_GLCOMD	WAIT_GDRDY-	FIFO_FULL	FIFO_EMPTY	VLD_RDY	GREQ	DMA_STATE[4:0]				
Type	RO	RO	RO			RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

DMA_STATE DMA FSM machine

- 0 IDLE
- 2 DMA_ADDR
- 4 DMA_RDATA
- 8 DMA_WAIT
- 16 DMA_PEND

GREQ DMA gmc request

VLD_RDY VLD codeword ready for CAVLC decoder

FIFO_EMPTY DMA FIFO empty

FIFO_FULL DMA FIFO full

WAIT_GDRDY Wait for gmc data ready signal

WAIT_GLCOMD Wait for gmc command latch

GADDR LSB-16 bit of DMA's requested memory address

H264+0104h H264 Decoder Debug INFO0 Register H264_DEC_DEBUG_INFO0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				SEQ_IDLE	VLD_ERR	VLD_RDY	GETBIT	BITCNT[4:0]					CAVLC_BLK_STATE[3:0]			
Type				RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CAVLC_MB_STATE[3:0]				CAVLC_Y_POS[5:0]						CAVLC_X_POS[5:0]					
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

CAVLC_MB_X Current macroblock X-position for CAVLC

CAVLC_MB_Y Current macroblock Y-position for CAVLC

CAVLC_MB_STATE CAVLC MB-level FSM state

- 0 MB_IDLE
- 1 RMEM_EN
- 2 RMEM

3 RMEM_DONE
4 WAIT_BLK_DONE
5 WMEM_EN
6 WMEM
7 WMEM_DONE
8 MB_DONE

CAVLC_BLK_STATE CAVLC 4x4 BLK-level FSM state

0 BLK_IDLE
1 RNC_WAIT
2 BLK_STR
3 TOTAL_COEFF
4 TRAIL1
5 LEVEL
6 TOTAL_ZERO
7 RUN
8 ERROR
9 NONCODED
10 BLK_DONE
11 LELEL_FULL
12 REST_COEFF

BITCNT Consumed bit counts for the current VLD-codeword
GETBIT Current VLD-codeword is valid
VLD_RDY VLD-codeword is ready for CAVLC
VLD_ERR VLD Error for the current frame.
SEQ_IDLE Sequencer is in IDLE state

H264+0108h H264 Decoder Debug INFO1 Register H264_DEC_DEBUG_INFO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VLD_CODEWORD[31:16]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLD_CODEWORD[15:0]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

VLD_CODEWORD Current VLD-codeword for CAVLC

H264+010Ch H264 Decoder Debug INFO2 Register H264_DEC_DEBUG_INFO2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							IDCT_OVERFLOW	BLOCK_4X4_CNT_IP[4:0]					BUSY[3:0]			
Type							RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TRANS[3:0]				INTRA/MC_Y_POS[5:0]						INTRA/MC_X_POS[5:0]					

Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

INTRA/MC_MB_X current macroblock X-position for Intra/MC

INTRA/MC_MB_Y current macroblock Y-position for Intra/MC

TRANS[3:0]

- **Bit0 : IS_INTRA4_TRANS** , MC state machine1 state
- **Bit1 : IS_INTRA16_TRANS** MC state machine2 state
- **Bit2 : IS_CHROMA_TRANS** MC state machine2 state
- **Bit3 : IS_PMB_TRANS** MC state machine2 state

BUSY[3:0]

- **Bit0 : IINTRA_BUSY**
- **Bit1 : IINTRA_4x4_MODE_BUSY**
- **Bit2 : MC_BUSY**
- **Bit3 : MV_BUSY**

BLOCK_4x4_CNT_IP

IDCT_OVERFLOW IDCT overflow for the current frame

H264+0110h H264 Decoder Debug INFO3 Register — H264_DEC_DEBUG_INFO3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								MVPRED_COUNTER[4:0]					MVPRED_STATE[3:0]			
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RI_STATE[1:0]		FIR_STATE_EXT[1:0]			FIR_STATE1[5:0]						MF_STATE[4:0]				
Type	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

MF_STATE Intra state machine1 state

FIR_STATE1 Intra state machine2 state

RI_STATE load_ri_base_addr_state

H264+0114h H264 Decoder Debug INFO4 Register — H264_DEC_DEBUG_INFO4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type															RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MC_PREFETCH_MEM_HIT_CNT[17:0]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

MC_PREFETCH_MEM_HIT_CNT MC pre-fetch memory hit counts information of current frame.

H264+0118h H264 Decoder Debug INFO5 Register — H264_DEC_DEBUG_INFO5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type															RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	MC_PREFETCH_MEM_MISS_CNT[17:0]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

MC_PREFETCH_MEM_MISS_CNT MC pre-fetch memory miss counts information of current frame.

H264+011Ch H264 Decoder Debug INFO6 Register H264_DEC_DEBUG_INFO6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						INTRA_GMC_STATE[4:0]					INTRA_STATE[5:0]					
Type						RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		ACC0[14:0]														
Type		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

INTRA_STATE Intra state machine state

H264+0120h H264 Decoder Debug INFO7 Register H264_DEC_DEBUG_INFO7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														INVTRANS_STATE[1:0]		QPY[5]
Type														RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	QPY[4:0]					INVQUANT_STATE[2:0]			INTRA4X4_GMC_STATE[2:0]			INTRA4X4_STATE[4:0]				
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

INTRA4x4_STATE Intra4x4 state machine

INTRA4x4_GMC_STATE Intra4x4 GMC controller state machine

H264+0124h H264 Decoder Debug INFO8 Register H264_DEC_DEBUG_INFO8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		MEM2_RDY	MEM1_RDY	FLR_RDY	DATA_BUF	DEBLK_FLR_STATE[2:0]			DEBLK_MEM2_STATE[3:0]				DEBLK_MEM1_STATE[3:0]			
Type		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEBLK_SEQ_STATE[3:0]				DEBLK_Y_POS[5:0]						DEBLK_X_POS[5:0]					
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

DEBLK_MB_X current macroblock X-position for deblocking filter

DEBLK_MB_Y current macroblock Y-position for deblocking filter

DEBLK_SEQ_STATE Deblocking filter's sequencer state machine state

- 0 IDLE
- 1 START
- 2 DEBLK_ST0
- 3 DEBLK_ST1
- 4 DEBLK_ST2
- 5 DEBLK_ST3

6 DEBLK_ST4
7 DEBLK_ST5
8 DEBLK_ST6
9 DEBLK_ST7
10 DEBLK_ST8

DEBLK_MEM1_STATE Deblocking filter's memory interface1 state machine state

0 IDLE
1 READ_REQ_1
2 READ_REQ_2
4 WRITE_REQ_1
5 WRITE_REQ_2
6 WRITE_REQ_3
8 WAIT_DAT_RDY1
9 WAIT_WRITE_OK1
10 WAIT_WRITE_OK2
11 WAIT_TO_IDLE

DEBLK_MEM2_STATE Deblocking filter's memory interface2 state machine state

0 IDLE
1 WAIT_BLK_DONE
4 WRITE_REQ_1
6 WRITE_REQ_2
9 WAIT_WRITE_OK1
10 WAIT_TO_IDLE

DEBLK_FLR_STATE Deblocking filter core state machine4 state

0 **IDLE**
1 **DEBLK_EDGE_0**
2 **DEBLK_EDGE_1**
3 **DEBLK_EDGE_2**
4 **DEBLK_EDGE_3**
5 **WAIT_DAT1**
6 **WAIT_DAT2**
7 **WAIT_EMPTY_BUFF**

DATA_BUF Current deblk filter used data buffer. **0: data buffer0; 1: data buffer1**

FLR_RDY Deblk filter core has finished in the current state

MEM1_RDY Deblk memory interface1 has finished in the current state

MEM2_RDY Deblk memory interface2 has finished in the current state

H264+0128h H264 Decoder Debug INFO9 Register H264_DEC_DEBUG_INFO9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	H264_DEC_CYCLE_CNT[31:16]															
Type		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	H264_DEC_CYCLE_CNT[15:0]															

© 2009 MediaTek Inc.

Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

H264_DEC_CYCLE_CNT Total cycle counts from H264_dec start to H264_dec_done per frame.

H264+012Ch H264 Decoder Debug INFO10 Register H264_DEC_DEBUG_INFO10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	H264_SEQ_STATE [47:32]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

H264_DEC_SEQ_STATE H264 Decoder state machine

H264+0130h H264 Decoder Debug INFO11 Register H264_DEC_DEBUG_INFO11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	H264_SEQ_STATE [31:16]															
Type		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	H264_SEQ_STATE[15:0]															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

H264_DEC_SEQ_STATE H264 Decoder state machine

5.16 Image DMA

5.16.1 General Description

Image DMA plays the role of moving image data between different image modules and memory. In MT6516, Image DMA has been divided into two modules, Image DMA 0 and Image DMA 1, and they are located in graphsys1 and graphsys2 respectively, as shown in **Figure 122**. In Image DMA 1, it mainly contains video and jpeg related sub-modules, and thus its major function is to provide services for Video Codec and JPEG Encoder. Besides, it contains two independent modules to support multiple outputs of OVL DMA and IRT0 DMA. In Image DMA 0, it contains several modules to support image frame buffer read/write. It also contains two extra modules to support multiple outputs of OVL DMA and IRT0 DMA.

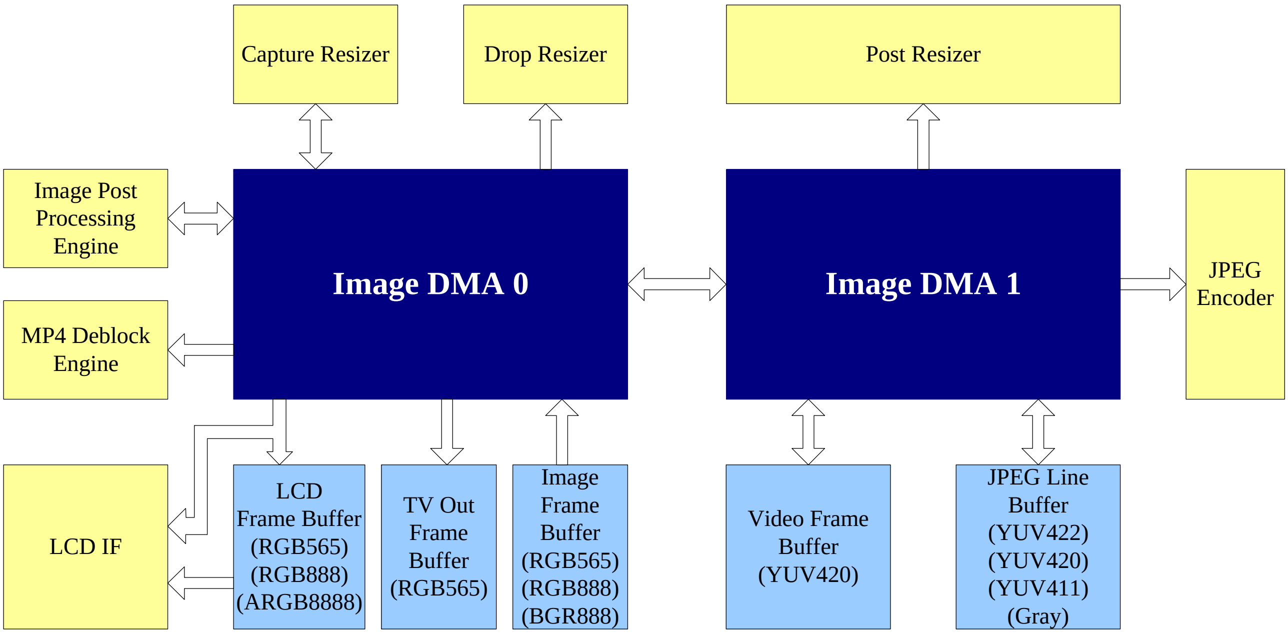


Figure 122 ***Inter-connection of Image DMA 0 and Image DMA 1***

5.16.1.1 Image DMA 0

Image DMA 0 contains 6 major modules. Most of them work in RGB domain except that OVL DMA works in YUV domain. They altogether support those functions listed below.

- Read RGB565/BGR888/RGB888 image frame and output by pixel.
- Overlay an image with a mask on-the-fly.
- Image clipping
- Image pitching
- Image rotation and flipping.
- Hardware triggering and direct couple interface to LCD DMA.

The details of each sub-module are described in the following.

5.16.1.1.1 Image Buffer Write 1 DMA (IBW1 DMA)

IBW1 DMA receives RGB888 pixel data from Y2R1 (YUV to RGV Engine 1), and either write data into memory or output pixel data to LCD. When writing to memory, it supports three formats, RGB565, RGB888 or ARGB8888, as shown in **Figure 123**. It plays the role of saving the backup image. Whenever JPEG DMA, Video DMA, or IRT1 DMA is dumping images, IBW1 DMA can be enabled to dump a backup image simultaneously. Besides, IBW1 DMA also plays the role of writing local display under videophone scenario. It has the following functions.

- Auto-restart and triple buffers supported
- Direct Couple to LCD DMA (DC mode)

- Hardware trigger to LCD DMA
- Image clipping
- Image pitching

Detail descriptions of those functions are described as below.

Auto-Restart

IBW1 DMA can restart itself to receive next frame, and switch base address at every restart. It has three base addresses to support double or triple frame buffers under auto-restart mode.

Direct Couple to LCD DMA (DC mode)

IBW1 DMA can move data to LCD DMA through direct couple interface. The interface consists of request, acknowledge, and 32-bit data bus. Under DC mode, frame data will skip the frame buffer, as indicated in **Figure 123**. LCD updates the data on the fly under this mode.

However this mode cannot work in camera preview. This is because LCD update could halt for a long time, and therefore the next pixel data from the camera may not be captured in time.

Hardware Trigger to LCD DMA

IBW1 DMA can issue a hardware trigger signal along with the current base address of the LCD frame buffer to acknowledge LCD DMA starting to move data. The signal would be either asserted at the start of a frame under DC mode or at the end of frame when non-DC mode, as shown in **Figure 124**.

Image Clipping

IBW1 DMA can grab a part of the input image frame as a new image, as illustrated in **Figure 125**. The advantage of image clipping is that the system does not need to prepare a large piece of memory to store the entire image frame just to show a small portion of it. This can save memory usage, especially for large images.

Image Pitching

IBW1 DMA can write the received image onto a background image existed in the memory already, as shown in **Figure 126**. In order to make the function work, two information must be provides for IBW1 DMA. The first is the base address where IBW1 would write the first pixel of the input image. The second is the horizontal size of the background image so that IBW1 DMA can get the starting address of the next line when it finishes a line. The pitching function can save memory usage when writing a rectangular area of a LCD layer.

 © 2009 MediaTek Inc.

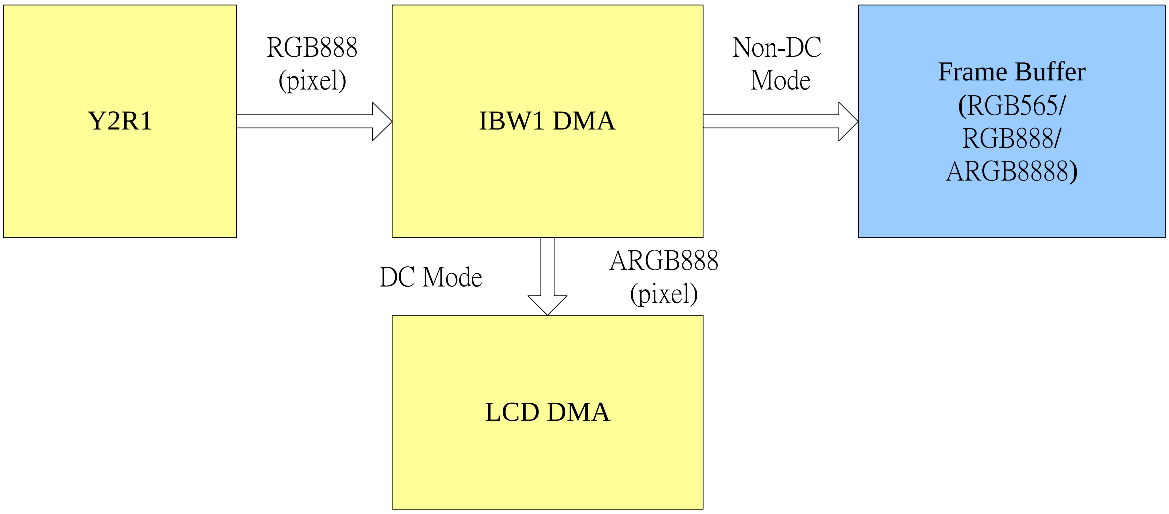


Figure 123 ***Inter-connection of IBW1 DMA***

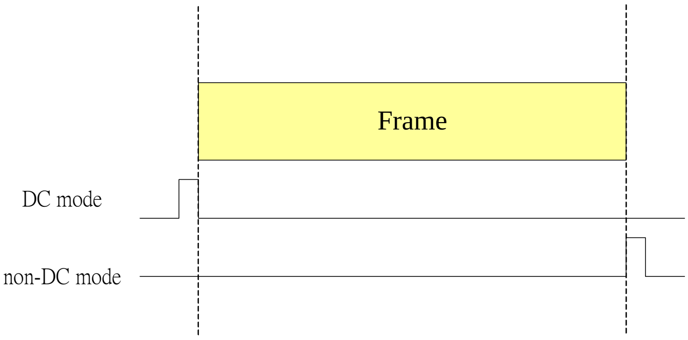


Figure 124 ***Hardware trigger signal of IBW1 DMA***

© 2009 MediaTek Inc.

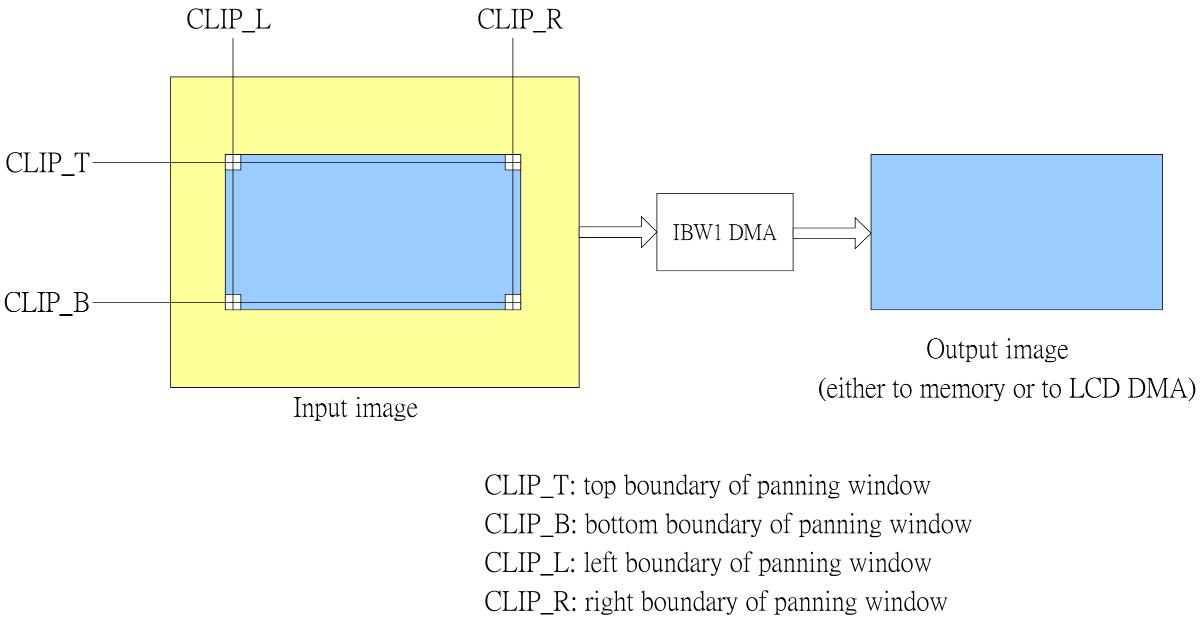


Figure 125 ***Image clipping of IBW1 DMA***

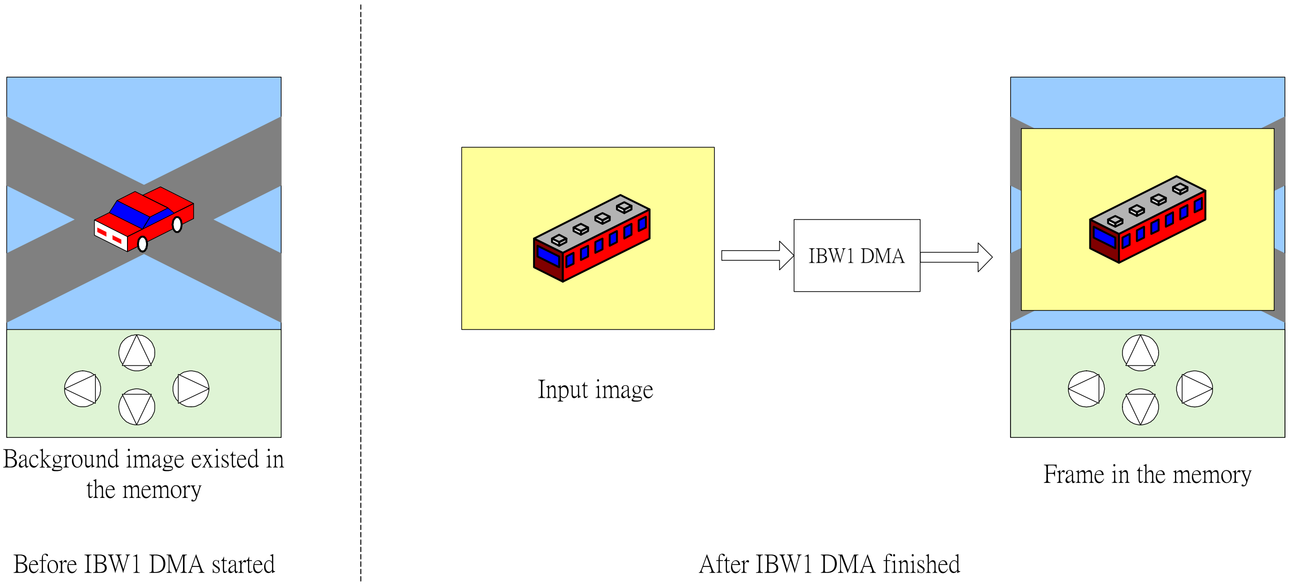


Figure 126 ***Image pitching of IBW1 DMA***

5.16.1.1.2 Image Buffer Write 2 DMA (IBW2 DMA)

IBW2 DMA receives RGB888 pixel data from IPP2 (Image Post Processor 2) and then output pixel data to LCD DMA, IRT1 DMA, or R2Y0 (RGB to YUV Engine 0). It is a pixel-in / pixel-out engine. Note that the three output paths could be enabled at the same time. Its functions are listed below.

- Auto-restart

- Direct Couple to LCD DMA (DC mode)
- Hardware trigger to LCD DMA
- Image clipping
- Multiple outputs

These first four functions are similar to those of IBW1 DMA except hardware trigger of IBW2 DMA. Because Direct Couple is the only way that IBW2 DMA handshakes with LCD DMA, IBW2 DMA issues a hardware trigger signal to LCD DMA only when the starting of the input image. For more details, please refer to the section **Image Buffer Write 1 DMA**. Here only the multiple output function is described.

Multiple outputs

IBW2 DMA supports multiple output function that it can output pixel data to IRT1 DMA, R2Y0, and LCD DMA simultaneously. This makes IDP paths more flexible.

5.16.1.1.3 Image Buffer Read 1 DMA (IBR1 DMA)

The main function of IBR1 DMA is to move RGB data from memory to R2Y0 (RBG to YUV Engine). The data format to R2Y0 is RGB888 and the data formats from memory can be RGB565, RGB888 and BGR888 (which support BMP data format). The data placement in memory is illustrated in **Figure 127**.

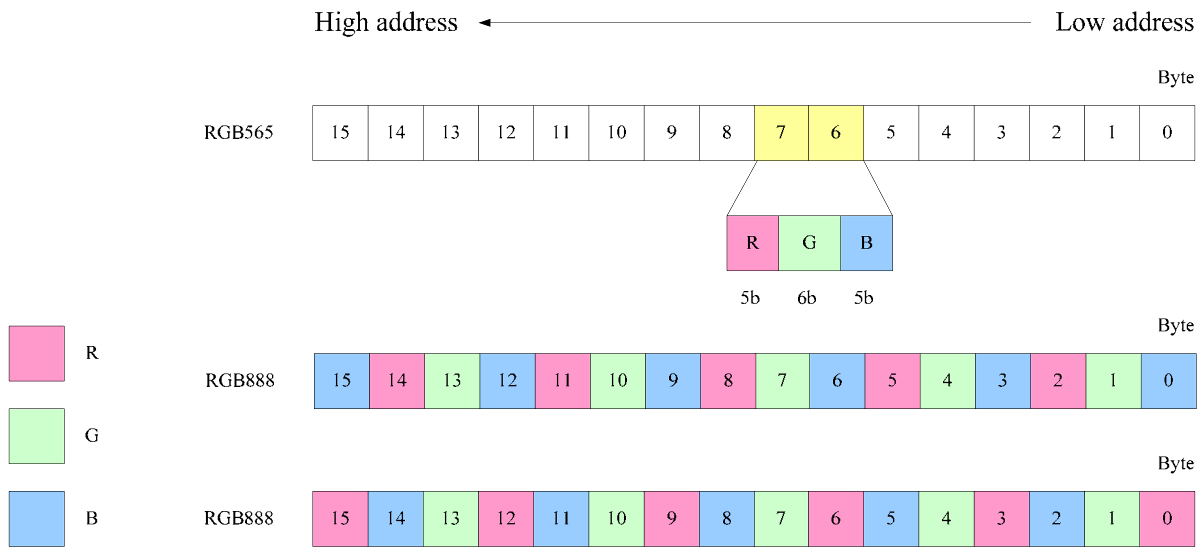


Figure 127 RGB data in memory for IBR1 DMA

5.16.1.1.4 Overlay DMA (OVL DMA) and OVL Multiple Output (OVL MO)

OVL DMA services several output engines, such as JPEG DMA, VDOENC DMA, PRZ (Post Resizer), DRZ (Drop Resizer), and Y2R0 (YUV to RGB Engine 0). The first three modules are located in graphsys2. In order to make parallel service for those engines possible, two multiple output engines are located in graphsys1 and graphsys2 respectively to achieve the goal, as shown in **Figure 128**.

The main function of OVL DMA is to read a photo frame from memory, magnify it to the capturing image size, look-up palette table and overlay it onto the capturing image, as shown in **Figure 129**. The photo frame mask data format can be 1, 2, 4, 8-bpp color index modes. A 256-entry in 24-bit YUV format palette lookup table is used to convert a color index to YUV color value.

© 2009 MediaTek Inc.

It is noted that the size of the input image should be multiple of the size of the mask. And although a 8-bits index is provided to select 256 colors, one value among 0~255 is used to be transparent color, i.e., if the value of the index is equal to color key, then it will output a pixel of the input image rather than of the mask.

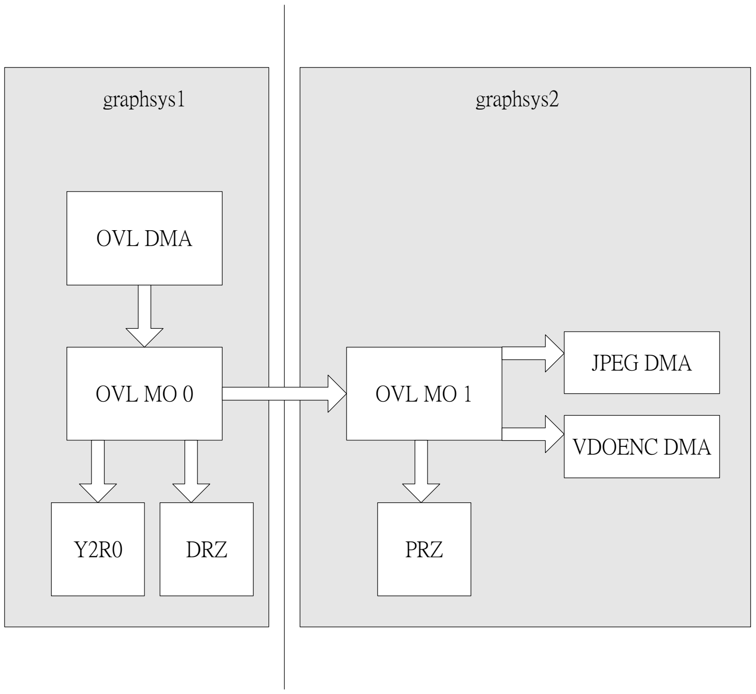


Figure 128 Multiple outputs of OVL DMA

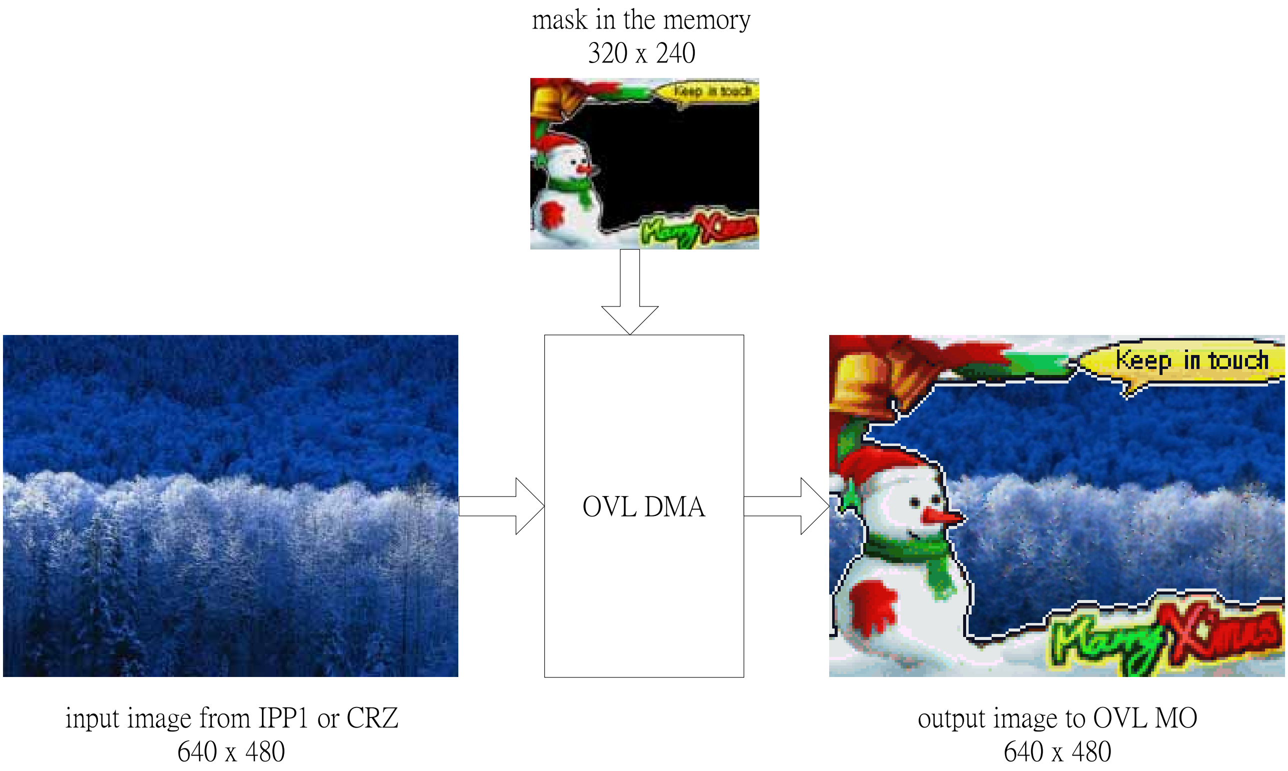


Figure 129 Overlay function of OVL DMA

5.16.1.1.5 Image Rotator 1 / Image Rotator 3 DMA (IRT1 / IRT3 DMA)

IRT1 and IRT3 DMA provide the same functions. They all received pixel data from the input image and write them into memory with RGB565, RGB888, or ARGB8888 format. The major functions of IRT1 / IRT3 DMA are listed below.

- Auto-restart and triple buffers supported
- Hardware trigger to LCD DMA (only IRT1 DMA)
- Image pitching
- Image rotation and flipping

The first three functions are similar to those of IBW1 DMA except hardware trigger of IRT1 DMA. Because IRT1 DMA has no Direct Couple to LCD DMA, the hardware trigger signal issued by IRT1 DMA always at the ending of the written image. For more details, please refer to the section **Image Buffer Write 1 DMA**, and here only the last function is described.

Image rotation and flipping

The definition of image rotation and flipping of IRT1 / IRT3 DMA is shown in **Figure 130**. When rotation function and flipping function are both enabled, the total effect is to rotate image first and then horizontally flip the rotated image, as indicated in the figure.

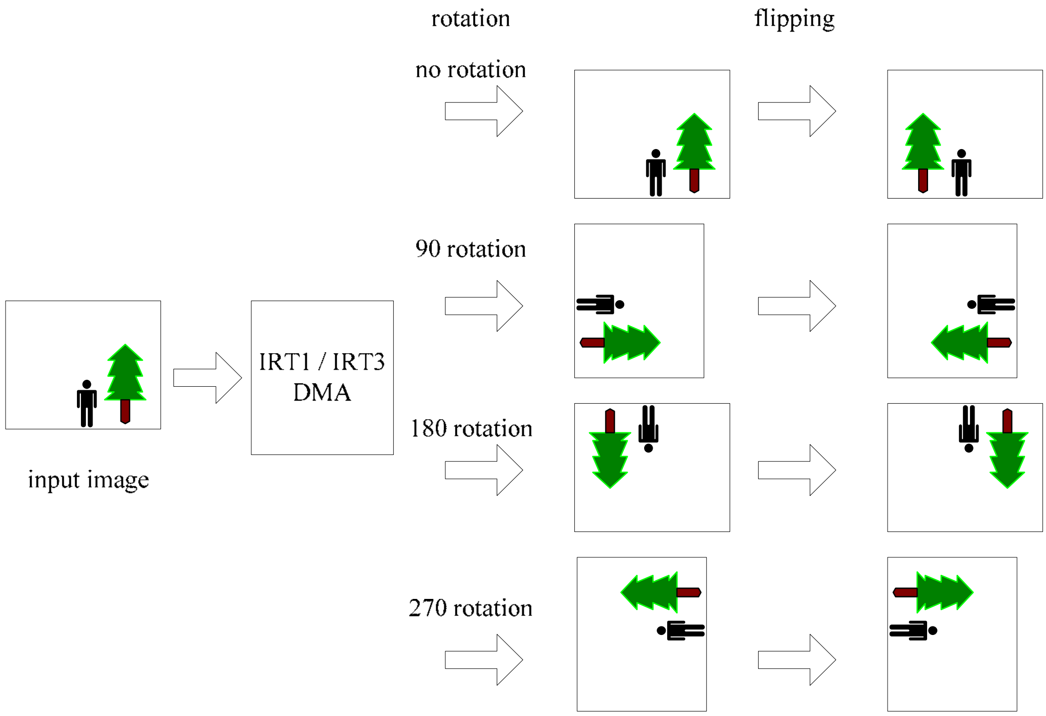


Figure 130 Definition of image rotation and flipping of IRT1 / IRT3 DMA

5.16.1.2 Image DMA 1

Image DMA 1 contains 4 major modules, and all of them work in YUV domain. They altogether support those functions listed below.

- Data stream flow control on JPEG Encoder DMA.
- Color format conversion (YUV444 □ YUV420, YUV444 □ YUV422, YUV444 □ YUV411)
- Data sequence conversion (scan-line based □ block based)
- Video rotation and flipping

The details of each sub-module are described in the following.

5.16.1.2.1 JPEG Encoder DMA (JPEG DMA)

The main function of JPEG DMA is to receive YUV444 data from OVL DMA by pixels and transmit YUV422/ YUV420/YUV411/gray data to JPEG Encoder by 8x8 blocks. Its main functions are described below.

- Auto-restart
- Color format conversion (YUV444 □ YUV420, YUV444 □ YUV422, YUV444 □ YUV411)
- Image padding
- Data format conversion
- Flow control

Detail descriptions of those functions are described as below.

Auto-Restart

To overlap the Codec processing time with that of file system manipulations, an Auto-Restart mode is designed. JPEG Encoder DMA automatically restarts itself to receive next frame without being re-configured and re-enabled by MCU. JPEG DMA will not stop transfer until it is disabled by MCU. Note that associated settings must be programmed in other related engines as well.

Color format conversion

JPEG DMA receives packed YUV444 data (pixel) from OVL DMA, and converts them into one of the four kinds of formats, i.e., YUV422, YUV420, YUV422, or gray data. The differences of these four formats are shortly described in **Table 105**.

	Y	U / V	
		Horizontal	Vertical
YUV422	No sub-sampled	Sub-sampled by 2	No sub-sampled
YUV420	No sub-sampled	Sub-sampled by 2	Sub-sampled by 2
YUV411	No sub-sampled	Sub-sampled by 4	No sub-sampled
Gray	No sub-sampled	Abandoned	Abandoned

Table 105 ***Format supported by JPEG DMA***

Image Padding

JPEG Encoder processes 8x8-block Y/U/V data, and JPEG DMA is responsible for transmitting these blocks. In order to ensure an image can be divided into 8x8 blocks in Y/U/V component respectively, JPEG DMA will pad the input image if needed. Referred to **Table 105**, if YUV422 format is selected, the U/V component of the input image will be horizontally sub-sampled by 2. To keep the sub-sampled component able to be divided into 8x8 blocks completely, the input image must be padded to be horizontally multiple of 16 and vertically multiple of 8 so that the size of U/V component after sub-sampled would be multiple of 8 in both horizontal and vertical. With the same consideration, the input image must be padded to multiple of 16 in both horizontal and vertical under YUV420 mode, multiple of 32 in horizontal and multiple of 8 under YUV411 mode, and multiple of 8 in both horizontal and vertical under gray mode. An example of YUV422 mode is illustrated in **Figure 131**. In this case, the original frame size is (16n+13) X (8n+6), which is not multiple of 16 X 8. Therefore, three additional pixels are padded to the end of each line to make the pixel count multiple of 16. In the vertical direction, two more lines are padded with last line to make line count to multiple of 8.

© 2009 MediaTek Inc.

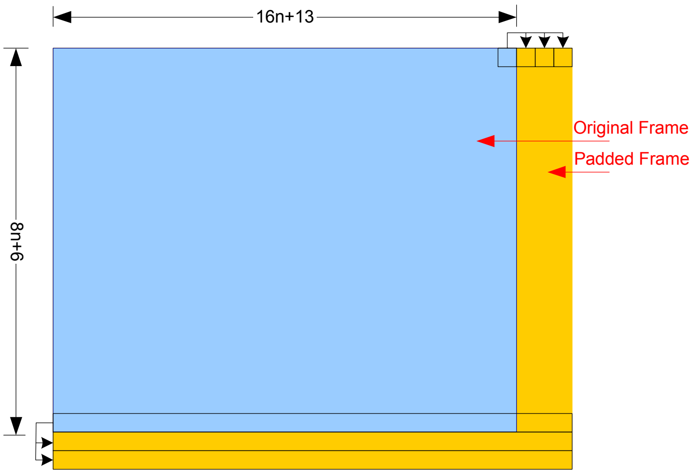


Figure 131 ***Image padding under YUV422 mode***

Data format conversion

Since the JPEG Encoder needs data of signed 2's complement format. The JPEG Encoder DMA takes the responsibility to handle the data format conversion. Each of Y, U, or V component values of input data is of 8-bit unsigned format, which represents a value ranging from 0 to 255. The component value of the data is converted into 8-bit 2's complement format, which represents a value ranging from –128 to 127.

Flow Control

When converting scan-line data into 8x8-block data, JPEG DMA uses line buffers to temporarily store the input pixel data. When working under YUV422, YUV411, and gray mode, JPEG DMA must store 8-line data and then it can transmit 8x8 blocks to JPEG Encoder. When working under YUV420 mode, because U/V component are vertically sub-sampled by 2, JPEG DMA must store 16-line data to make U/V component be divided into 8x8 blocks completely. After JPEG DMA collects enough line data, it transmits blocks inside a macro block to JPEG Encoder. After a macro-block finishes, it continue to transmit blocks inside the next macro block. The blocks inside a macro-block is determined by its format, as shown in **Figure 132**. Note that the number inside each macro block in **Figure 132** represents the sequence of Y blocks.

 © 2009 MediaTek Inc.

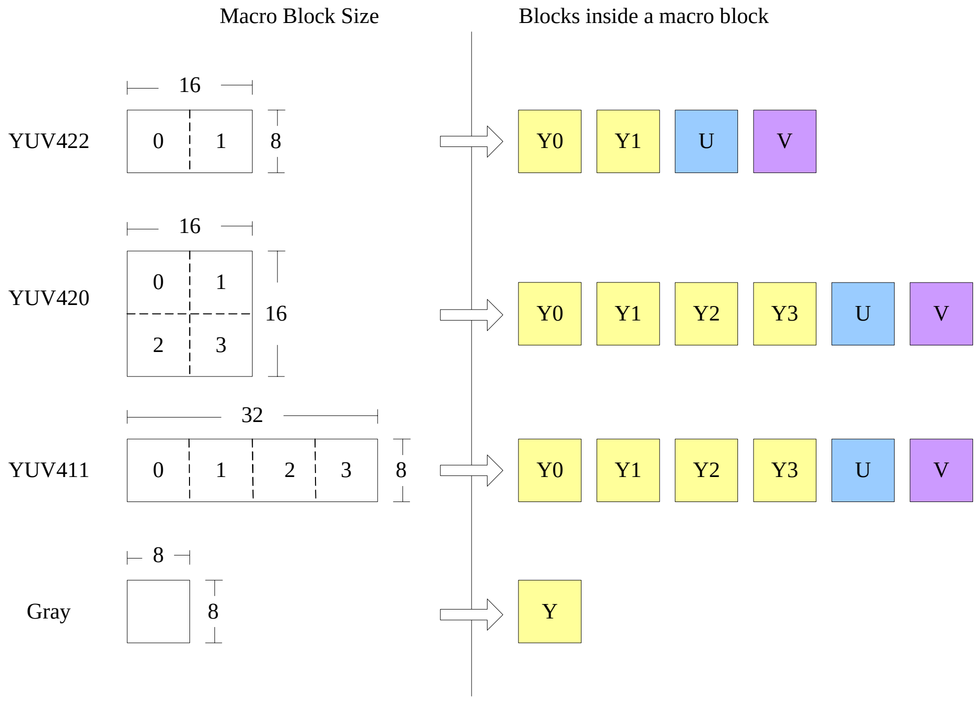


Figure 132 ***Sequence of blocks transmitted to JPEG Encoder***

5.16.1.2.2 Video Encode DMA (VDOENC DMA)

The main functions of VDOENC DMA are listed below.

- Auto-restart
- Color format conversion (packed YUV444 data planar YUV 420 data)
- Co-work with IRT0 DMA to support video frame rotation and flipping

VDOENC DMA has a frame buffer write engine (VDOENC WDMA) and a frame buffer read engine (VDOENC RDMA) to handle frame buffers, as shown in **Figure 133** . VDOENC DMA is responsible for converting packed YUV444 data (pixel) to planar YUV420 data, and VDOENC WDMA writes those planar data into three contiguous component buffers (Y/U/V buffer) for MPEG4 Encoder. For some design consideration, the data sequece of each component is 4x4 block now, and thus VDOENC DMA needs line buffers to convert the scan-line data into 4x4 block data.

By the way, for video preview, VDOENC RDMA can be triggered by VDOENC WDMA to reads the planar YUV420 data buffers written by VDOENC WDMA, and then output to Image Rotator 0 DMA (IRT0 DMA). If VDOENC RDMA is not idle when VDOENC WDMA issues a trigger signal, it will ignore the signal and continue its current transmission.

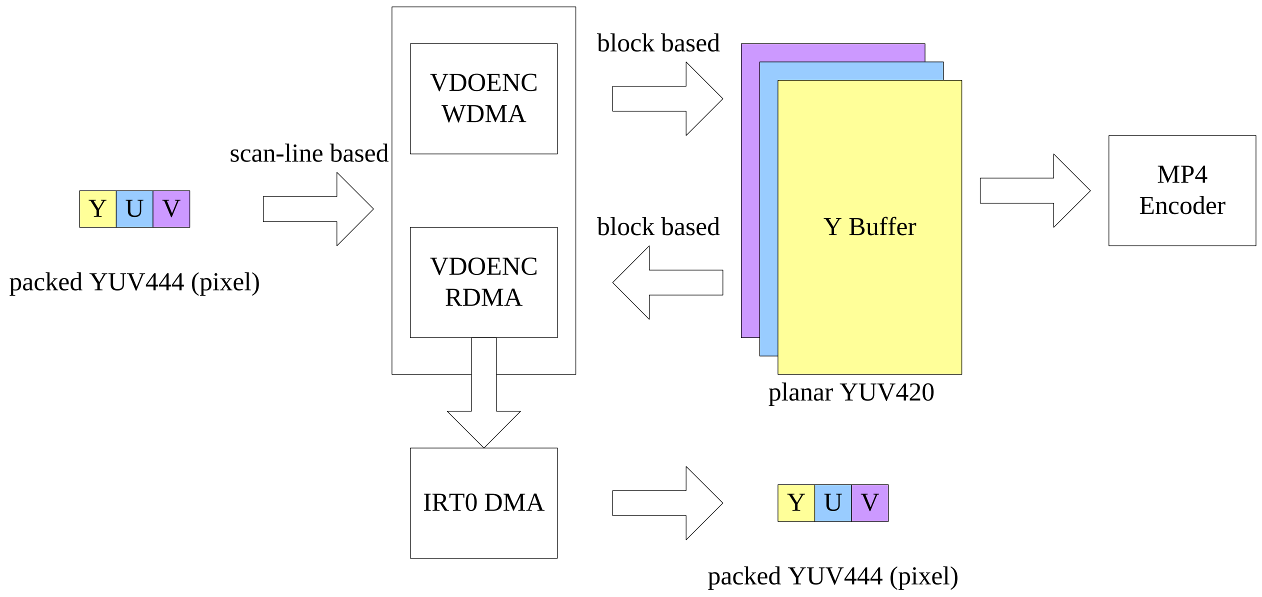


Figure 133 ***Overview of VDOENC DMA***

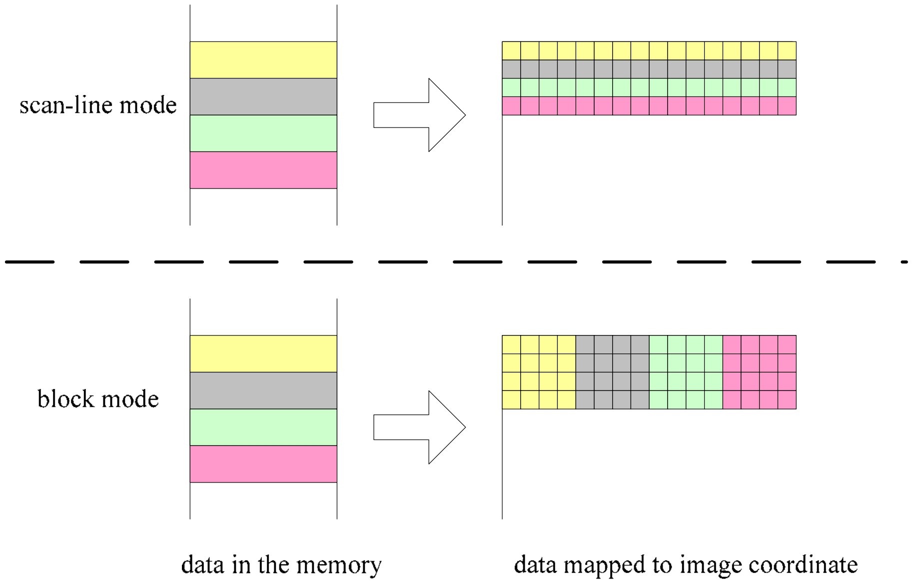


Figure 134 ***Scan-line mode vs. block mode***

Auto-Restart

VDOECN WDMA automatically restarts itself to receive next frame without being re-configured and re-enabled by MCU. This can save a lot of MCU time since the MCU no longer needs to handle VDOENC DMA at each frame boundary, which also makes the data stream smoother. Usually, double buffer scheme is employed to smooth video encoding. Therefore, the second set of base addresses register are provided in VDOENC DMA to contain the second frame buffer. VDOENC WDMA automatically switches the base address

between the two addresses at every restart. By the way, as mentioned above, VDOENC RDMA is triggered by VDOENC WDMA, so it can be restarted by VDOENC WDMA when it is idle. The base address of VDOENC RDMA is sent by VDOENC WDMA to ensure VDOENC RDMA reads the frame buffer which VDOENC WDMA just wrote.

Color format conversion

As mentioned, VDOENC DMA converts packed YUV444 data into planar YUV420 data. Besides, it also converts the scan-line data into 4x4 block data.

Rotation and flipping

VDOECN RDMA must co-work with IRT0 DMA to execute the functions. The data forward to IRT0 DMA are 4x4 block data, and IRT0 will convert them into packed YUV444 data. The definitions of rotation and flipping are the same to those of IRT1 DMA. Please refer to **Figure 130** for more details.

5.16.1.2.3 Video Decode DMA (VDODEC DMA)

The main functions of VDODEC DMA are listed below.

- Both scan-line based buffer and block-based buffer supported
- Color format conversion (planar YUV 420 data □ packed YUV444 data)
- Co-work with IRT0 DMA to support video frame rotation and flipping

Scan-line mode vs. block mode

VDODEC can read 4x4 block based frame buffer and scan-line based frame buffer, as shown in **Figure 135**. When VDODEC DMA works under scan-line mode, it will output to IRT0 Multiple Output Engine directly, and thus unable to rotate or flip the frame.

Color format conversion

When working under scan-line mode, VDODEC will output packed YUV444 data to the succeeding module. Thus conversion from planar YUV420 to pack YUV444 is done inside VDODEC DMA.

Co-work with IRT0 DMA

When working under block mode, VDODEC DMA can co-work with IRT0 DMA to execute the rotation and flipping functions, just as VDOENC RDMA does.

 © 2009 MediaTek Inc.

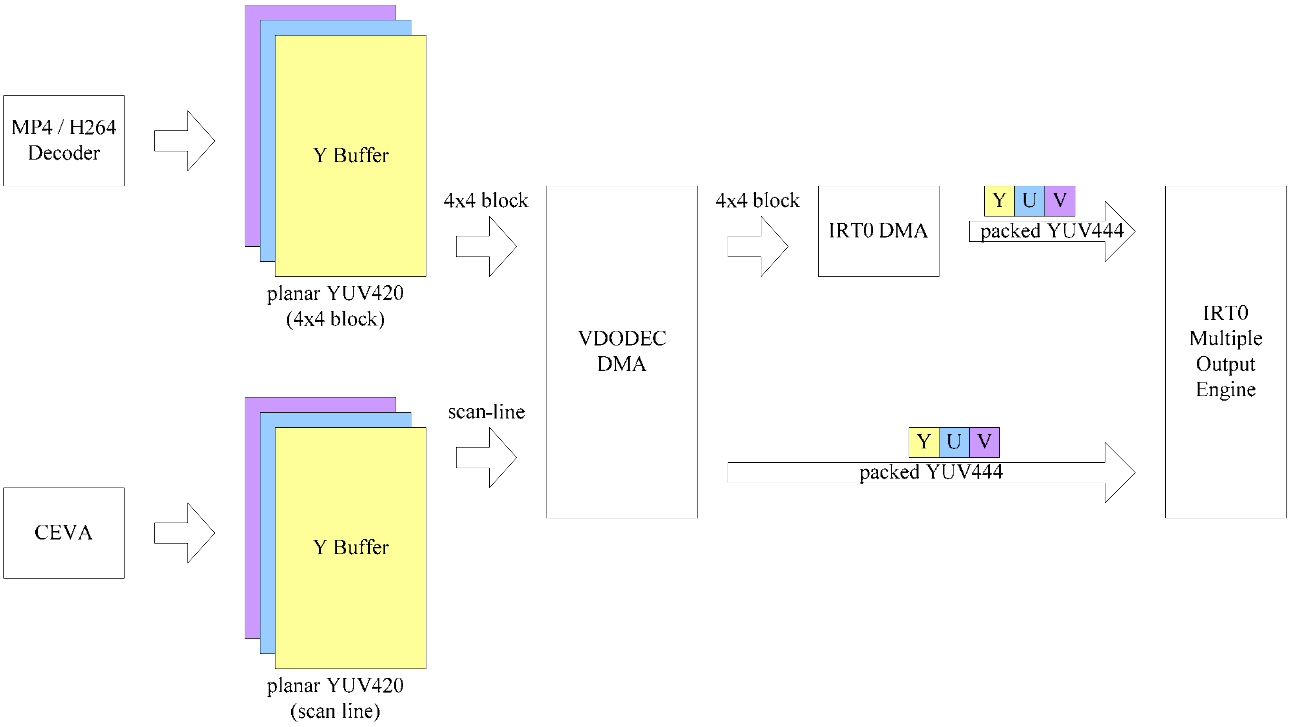


Figure 135 ***Overview of VDODEC DMA***

5.16.1.2.4 Image Rotator 0 DMA (IRT0 DMA) and IRT0 Multiple Output (IRT0 MO)

IRT0 DMA has three main functions.

- Convert 4x4 block data to packed YUV444 data
- Co-work with Video Encode RDMA or Video Decode DMA to support video frame rotation and flipping
- Multiple outputs

IRT0 DMA uses line buffers to complete the functions above. About the definition of rotation and flipping, please refer to **Figure 130**. Here only the function of multiple outputs is described.

Multiple outputs of IRT0 DMA

IRT0 DMA services several output engines, such as Capture Resize (CRZ), Image Post Processor 1 (IPP1), MPEG4 de-blocking Engine (MP4DEBLK), and Post Resizer (PRZ). The first three modules are located in graphsys1. In order to make parallel service for those engines possible, two multiple output engines are located in graphsys1 and graphsys2 respectively to achieve the goal, as shown in **Figure 136**.

 © 2009 MediaTek Inc.

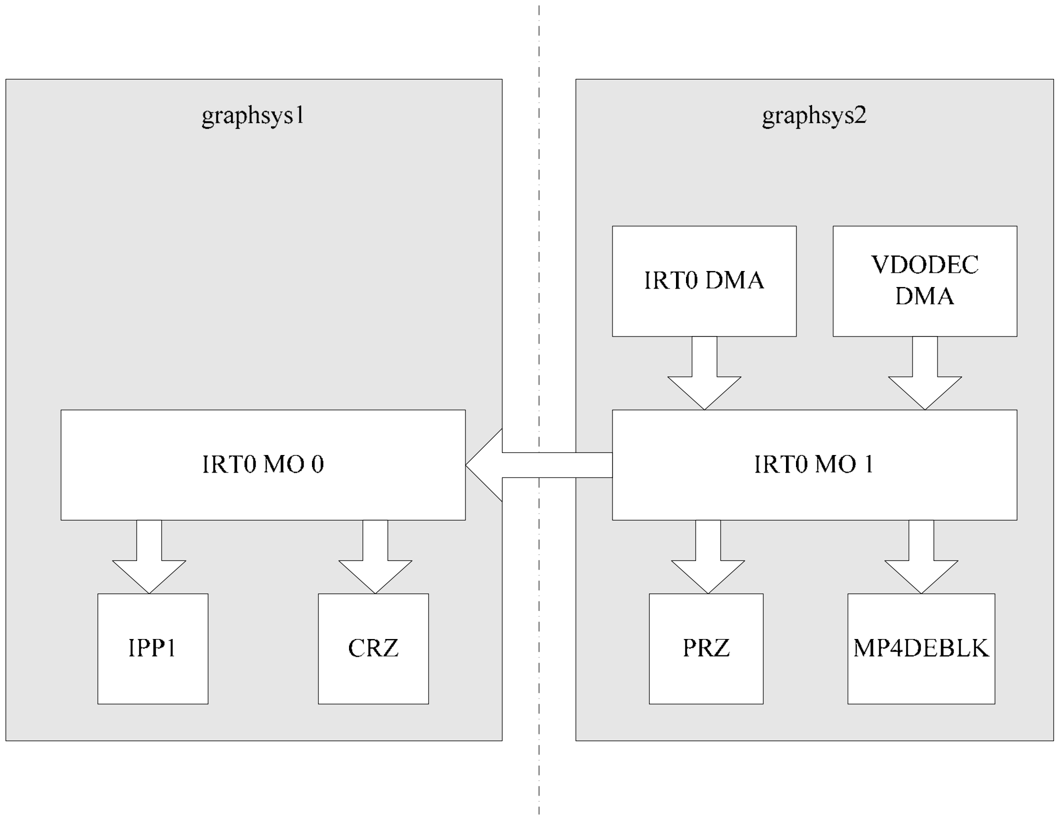


Figure 136 Multiple outputs of IRT0 DMA

 © 2009 MediaTek Inc.

5.16.2 Register Definitions

Engine	Width/Height	Bits	Minimum	Maximum Value
IBW1	SRC_XSIZE	16	1	65,535
	SRC_YSIZE	16	1	65,535
	BKGD_XSIZE0	16	1	65,535
	BKGD_XSIZE1	16	1	65,535
	BKGD_XSIZE2	16	1	65,535
IBW2	XSIZE	16	1	65,535
	YSIZE	16	1	65,535
IBR1	PXLNUM	32	1	4,294,967,295
OVL	XSIZE	16	1	65,535
	YSIZE	16	1	65,535
IRT1	SRC_XSIZE	16	1	65,535
	SRC_YSIZE	16	1	65,535
	BKGD_XSIZE0	16	1	65,535
	BKGD_XSIZE1	16	1	65,535
	BKGD_XSIZE2	16	1	65,535
IRT3	SRC_XSIZE	16	1	65,535
	SRC_YSIZE	16	1	65,535
	BKGD_XSIZE0	16	1	65,535
	BKGD_XSIZE1	16	1	65,535
	BKGD_XSIZE2	16	1	65,535
JPEG	XSIZE	16	1	65,535
	YSIZE	16	1	65,535
VDOENC	XSIZE	12	16 (16x)	4080 (16x)
	YSIZE	12	16 (16x)	4080 (16x)

© 2009 MediaTek Inc.

VDODEC	XSIZE	10	16 (16x)	1008 (16x)
[	YSIZE	10		
ľ			16 (16x)	1008 (16x)
/				
IRT0	XSIZE	12	16 (16x)	4080 (16x)
[	YSIZE	12		
ľ			16 (16x)	4080 (16x)
/				

5.16.2.1 Image DMA 0

Register Address	Register Function	Acronym
IMGDMA0+0000h	Image DMA 0 Status Register	IMGDMA0_STA
IMGDMA0+0004h	Image DMA 0 Interrupt Acknowledge Register	IMGDMA0_ACKINT
IMGDMA0+0010h	Image DMA Software Reset	IMGDMA0_SW_RSTB
IMGDMA0+0020h	Image DMA 0 GMC Interface Status	IMGDMA0_GMCIF_STA
IMGDMA0+0040h	Image DMA 0 Current Frame Information	IMGDMA0_CURR_FRAME
IMGDMA0+0300h	Image Buffer Write 1 DMA Start Register	IMGDMA0_IBW1_STR
IMGDMA0+0304h	Image Buffer Write 1 DMA Control Register	IMGDMA0_IBW1_CON
IMGDMA0+0308h	Image Buffer Write 1 DMA Alpha Register	IMGDMA0_IBW1_ALPHA
IMGDMA0+0310h	Image Buffer Write 1 DMA Horizontal Size Register of Source Image	IMGDMA0_IBW1_SRC_XSIZE
IMGDMA0+0314h	Image Buffer Write 1 DMA Vertical Size Register of Source Image	IMGDMA0_IBW1_SRC_YSIZE
IMGDMA0+0318h	Image Buffer Write 1 DMA CLIP Left/Right Coordinate Register of Source Image	IMGDMA0_IBW1_CLIPLR
IMGDMA0+031Ch	Image Buffer Write 1 DMA CLIP Top/Bottom Coordinate Register of Source Image	IMGDMA0_IBW1_CLIPTB
IMGDMA0+0320h	Image Buffer Write 1 DMA Base Address Register of Frame Buffer 0	IMGDMA0_IBW1_BASE_ADDR0
IMGDMA0+0324h	Image Buffer Write 1 DMA Base Address Register of Frame Buffer 1	IMGDMA0_IBW1_BASE_ADDR1
IMGDMA0+0328h	Image Buffer Write 1 DMA Base	IMGDMA0_IBW1_BASE_ADDR2

© 2009 MediaTek Inc.

	Address Register of Frame Buffer 2	
IMGDMA0+0330h	Image Buffer Write 1 DMA Horizontal Size Register of Background Image 0	IMGDMA0_IBW1_BKGD_XSIZE0
IMGDMA0+0334h	Image Buffer Write 1 DMA Horizontal Size Register of Background Image 1	IMGDMA0_IBW1_BKGD_XSIZE1
IMGDMA0+0338h	Image Buffer Write 1 DMA Horizontal Size Register of Background Image 2	IMGDMA0_IBW1_BKGD_XSIZE2
IMGDMA0+0340h	Image Buffer Write 1 DMA Horizontal Pixel Count Register of Received Image	IMGDMA0_IBW1_RX_XCNT
IMGDMA0+0344h	Image Buffer Write 1 DMA Vertical Line Count Register of Received Image	IMGDMA0_IBW1_RX_YCNT
IMGDMA0+0350h	Image Buffer Write 1 DMA Horizontal Byte Count Register of Written Image	IMGDMA0_IBW1_HORI_CNT
IMGDMA0+0354h	Image Buffer Write 1 DMA Vertical Line Count Register of Written Image	IMGDMA0_IBW1_VERT_CNT
IMGDMA0+0400h	Image Buffer Write 2 DMA Start Register	IMGDMA0_IBW2_STR
IMGDMA0+0404h	Image Buffer Write 2 DMA Control Register	IMGDMA0_IBW2_CON
IMGDMA0+0408h	Image Buffer Write 2 DMA Alpha Value Register	IMGDMA0_IBW2_ALPHA
IMGDMA0+0410h	Image Buffer Write 2 DMA Horizontal Size Register of Source Image	IMGDMA0_IBW2_XSIZE
IMGDMA0+0414h	Image Buffer Write 2 DMA Vertical Size Register of Source Image	IMGDMA0_IBW2_YSIZE
IMGDMA0+0418h	Image Buffer Write 2 DMA CLIP Left/Right Coordinate Register of Source Image	IMGDMA0_IBW2_CLIPLR
IMGDMA0+041Ch	Image Buffer Write 2 DMA CLIP Top/Bottom Coordinate Register of Source Image	IMGDMA0_IBW2_CLIPTB
IMGDMA0+0420h	Image Buffer Write 2 DMA Horizontal Pixel Count Register of Received Image	IMGDMA0_IBW2_XCNT
IMGDMA0+0424h	Image Buffer Write 2 DMA Vertical Line Count Register of Received Image	IMGDMA0_IBW2_YCNT

 © 2009 MediaTek Inc.

IMGDMA0+0500h	Image Buffer Read 1 DMA Start Register	IMGDMA0_IBR1_STR
IMGDMA0+0504h	Image Buffer Read 1 DMA Control Register	IMGDMA0_IBR1_CON
IMGDMA0+0508h	Image Buffer Read 1 DMA Base Address Register of Source Image	IMGDMA0_IBR1_BASE
IMGDMA0+050Ch	Image Buffer Read 1 DMA Pixel Number Register of Source Image	IMGDMA0_IBR1_PXLNUM
IMGDMA0+0510h	Image Buffer Read 1 DMA Pixel Count Register of Source Image	IMGDMA0_IBR1_PXLCNT
IMGDMA0+0700h	Overlay DMA Start Register	IMGDMA0_OVL_STR
IMGDMA0+0704h	Overlay DMA Control Register	IMGDMA0_OVL_CON
IMGDMA0+0708h	Overlay DMA Base Address Register of Mask Image	IMGDMA0_OVL_BASE
IMGDMA0+070Ch	Overlay DMA Configuration Register	IMGDMA0_OVL_CFG
IMGDMA0+0710h	Overlay DMA Horizontal Size Register of Mask Image	IMGDMA0_OVL_XSIZE
IMGDMA0+0714h	Overlay DMA Vertical Size Register of Mask Image	IMGDMA0_OVL_YSIZE
IMGDMA0+0718h	Overlay DMA Horizontal Pixel Count Register of Mask Image	IMGDMA0_OVL_XCNT
IMGDMA0+071Ch	Overlay DMA Vertical Line Count Register of Mask Image	IMGDMA0_OVL_YCNT
IMGDMA0+0800h	Overlay DMA Palette Register 00	IMGDMA0_OVL_PAL_BASE
IMGDMA0+0780h	Overlay DMA Multiple Output Engine 0 Start Register	IMGDMA0_OVL_MO_0_STR
IMGDMA0+0784h	Overlay DMA Multiple Output Engine 0 Control Register	IMGDMA0_OVL_MO_0_CON
IMGDMA0+0788h	Overlay DMA Multiple Output Engine 0 Busy Status	IMGDMA0_OVL_MO_0_BUSY
IMGDMA0+0C80h	Image Rotator 0 DMA Multiple Output Engine 0 Start Register	IMGDMA0_IRT0_MO_0_STR
IMGDMA0+0C84h	Image Rotator 0 DMA Multiple Output Engine 0 Control Register	IMGDMA0_IRT0_MO_0_CON
IMGDMA0+0C88h	Image Rotator 0 DMA Multiple Output Engine 0 Busy Status	IMGDMA0_IRT0_MO_0_Busy
IMGDMA0+0D00h	Image Rotator 1 DMA Start Register	IMGDMA0_IRT1_STR
IMGDMA0+0D04h	Image Rotator 1 DMA Control Register	IMGDMA0_IRT1_CON

IMGDMA0+0D08h	Image Rotator 1 DMA Alpha Value Register	IMGDMA0_IRT1_ALPHA
IMGDMA0+0D10h	Image Rotator 1 DMA Horizontal Size Register of Source Image	IMGDMA0_IRT1_SRC_XSIZE
IMGDMA0+0D14h	Image Rotator 1 DMA Vertical Size Register of Source Image	IMGDMA0_IRT1_SRC_YSIZE
IMGDMA0+0D20h	Image Rotator 1 DMA Base Address of Frame Buffer 0	IMGDMA0_IRT1_BASE_ADDR0
IMGDMA0+0D24h	Image Rotator 1 DMA Base Address of Frame Buffer 1	IMGDMA0_IRT1_BASE_ADDR1
IMGDMA0+0D28h	Image Rotator 1 DMA Base Address of Frame Buffer 2	IMGDMA0_IRT1_BASE_ADDR2
IMGDMA0+0D30h	Image Rotator 1 DMA Horizontal Size Register of Background Image 0	IMGDMA0_IRT1_BKGD_XSIZE0
IMGDMA0+0D34h	Image Rotator 1 DMA Horizontal Size Register of Background Image 1	IMGDMA0_IRT1_BKGD_XSIZE1
IMGDMA0+0D38h	Image Rotator 1 DMA Horizontal Size Register of Background Image 2	IMGDMA0_IRT1_BKGD_XSIZE2
IMGDMA0+0D40h	Image Rotator 1 DMA Base Address Register of Line Buffer	IMGDMA0_IRT1_FIFO_BASE
IMGDMA0+0D50h	Image Rotator 1 DMA Horizontal Pixel Count Register of Received Image	IMGDMA0_IRT1_RX_XCNT
IMGDMA0+0D54h	Image Rotator 1 DMA Vertical Line Count Register of Received Image	IMGDMA0_IRT1_RX_YCNT
IMGDMA0+0D60h	Image Rotator 1 DMA Horizontal Byte Count Register of Written Image	IMGDMA0_IRT1_HORI_CNT
IMGDMA0+0D64h	Image Rotator 1 DMA Vertical Line Count Register of Written Image	IMGDMA0_IRT1_VERT_CNT
IMGDMA0+0F00h	Image Rotator 3 DMA Start Register	IMGDMA0_IRT3_STR
IMGDMA0+0F04h	Image Rotator 3 DMA Control Register	IMGDMA0_IRT3_CON
IMGDMA0+0F08h	Image Rotator 3 DMA Alpha Value Register	IMGDMA0_IRT3_ALPHA
IMGDMA0+0F10h	Image Rotator 3 DMA Horizontal Size Register of Source Image	IMGDMA0_IRT3_SRC_XSIZE
IMGDMA0+0F14h	Image Rotator 3 DMA Vertical Size Register of Source Image	IMGDMA0_IRT3_SRC_YSIZE

 © 2009 MediaTek Inc.

IMGDMA0+0F20h	Image Rotator 3 DMA Base Address of Frame Buffer 0	IMGDMA0_IRT3_BASE_ADDR0
IMGDMA0+0F24h	Image Rotator 3 DMA Base Address of Frame Buffer 1	IMGDMA0_IRT3_BASE_ADDR1
IMGDMA0+0F28h	Image Rotator 3 DMA Base Address of Frame Buffer 2	IMGDMA0_IRT3_BASE_ADDR2
IMGDMA0+0F30h	Image Rotator 3 DMA Horizontal Size Register of Background Image 0	IMGDMA0_IRT3_BKGD_XSIZE0
IMGDMA0+0F34h	Image Rotator 3 DMA Horizontal Size Register of Background Image 1	IMGDMA0_IRT3_BKGD_XSIZE1
IMGDMA0+0F38h	Image Rotator 3 DMA Horizontal Size Register of Background Image 2	IMGDMA0_IRT3_BKGD_XSIZE2
IMGDMA0+0F40h	Image Rotator 3 DMA Base Address Register of Line FIFO	IMGDMA0_IRT3_FIFO_BASE
IMGDMA0+0F50h	Image Rotator 3 DMA Horizontal Pixel Count Register of Received Image	IMGDMA0_IRT3_RX_XCNT
IMGDMA0+0F54h	Image Rotator 3 DMA Vertical Line Count Register of Received Image	IMGDMA0_IRT3_RX_YCNT
IMGDMA0+0F60h	Image Rotator 3 DMA Horizontal Byte Count Register of Written Image	IMGDMA0_IRT3_HORI_CNT
IMGDMA0+0F64h	Image Rotator 3 DMA Vertical Line Count Register of Written Image	IMGDMA0_IRT3_VERT_CNT

Table 106 ***Image DMA 0 Register Map***

IMGDMA0+0000h Image DMA 0 Status Register IMGDMA0_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRT3 RUN**		**IRT1 RUN**			**OVL RUN**		**IBR1 RUN**			**IBW2 RUN**	**IBW1 RUN**				
Type	RO		RO			RO		RO			RO	RO				
Reset	0		0			0		0			0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IRT3 IT**		**IRT1 IT**			**OVL IT**		**IBR1 IT**			**IBW2 IT**	**IBW1 IT**				
Type	RO		RO			RO		RO			RO	RO				
Reset	0		0			0		0			0	0				

This register helps software program being well aware of the global status of Image DMA 0 channels.

IT:

Interrupt status for engines inner Image DMA 0

0: No interrupt is generated by associated engine of Image DMA 0

1: An interrupt is generated by associated engine of Image DMA 0 and waiting for service. It would be cleared after setting corresponding bit in IMGDMA0_ACKINT Register.

RUN:

DMA engine status

0: DMA engine is stopped or has completed the transfer already.

1: DMA engine is currently running. If the engine is set to be auto-restart, the related bits will be always 1 until the engine is turned off.

IMGDMA0+0004h Image DMA 0 Interrupt Acknowledge Register IMGDMA0_ACKINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRT3 ACK		IRT1 ACK			OVL ACK		IBR1 ACK			IBW2 ACK	IBW1 ACK				
Type	WO		WO			WO		WO			WO	WO				

This register is used to acknowledge the current interrupt request associated with the completion event of a DMA channel by software program. Note that this is a write-only register, and any read to it will return a value of “0”.

ACK:

Interrupt acknowledge for the DMA channel

0: No effect

1: Interrupt request of associated engine is acknowledged and interrupt status shown in IMGDMA0_STA should be relinquished.

IMGDMA0+0010h Image DMA Software Reset IMGDMA0_SW_RSTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																RSTB
Type																RW
Reset																1

This register is used to reset all registers inner sub-engines except those which are setting through APB. ***The start bit of each engine should be set to 0 before setting this bit to 0. Once Image DMA is reset, it should be restarted after at least 100us.***

RSTB:

Software reset bit.

0: Reset all registers inner sub-engines of Image DMA 0 and Image DMA 1 except those which are setting through APB.

1: No effect.

IMGDMA0+0020h Image DMA 0 GMC Interface Status IMGDMA0_GMCIF_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								IRT3 RINT GMC BUSY	IRT1 RINT GMC BUSY	IRT3 WINT GMC BUSY	IRT1 WINT GMC BUSY	OVL GMC BUSY	IBW1 GMC BUSY	IBR1 GMC BUSY	IRT3 WEXT GMC BUSY	IRT1 WEXT GMC BUSY
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												IF4 BUSY	IF3 BUSY	IF2 BUSY	IF1 BUSY	IF0 BUSY
Type												RO	RO	RO	RO	RO

There are 9 GMC related engines and five GMC ports in Image DMA 0. This register is used to figure out which engine or which set of GMC interface is busy. When the engine or GMC interface is busy, it means that there are data waiting to read from / write into memory.

XXX GMC BUSY:

Engine busy signal.

0: Engine does not issue a command.

1: Engine issues a command and not completed yet.

IFX BUSY:

GMC Interface busy signal.

0: GMC interface is idle. No action occurs in this port.

1: GMC interface is busy. A command is not completed yet.

IMGDMA0+0040h Image DMA Current Frame Information IMGDMA0_CURR_FRAME

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRT1_CUR_FRAME		IBW1_CUR_FRAME													
Type	RO		RO													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																

IBW1_CUR_FRMAE:

To represent the current working frame of IBW1 DMA.

IRT1_CUR_FRMAE:

To represent the current working frame of IRT1 DMA.

IMGDMA0+0300h **Image Buffer Write 1 DMA Start Register** **IMGDMA0_IBW1_STR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IBW1 DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IBW1 DMA.

0: stop IBW1 DMA

1: activate IBW1 DMA

IMGDMA0+0304h **Image Buffer Write 1 DMA Control Register** **IMGDMA0_IBW1_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**FMT**		**DC**	**LCD**	**CLIP**	**PITCH**		**TRIPLE**	**AUTO RSTR**	**IT**
Type							R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset							0		0	0	0	0		0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. IBW1 DMA automatically restarts itself while current frame is finished.

0: Disable

1: Enable

TRIPLE:

When automatic restart function is enabled, it could enable this bit to support triple frame buffers; else, two frame

buffers are supported under auto-restart mode.

0: Two Frame Buffer Supported

1: Three Frame Buffer Supported

PITCH:

Destination pitching. Please refer to **Figure 126**.
0: Disable
1: Enable

CLIP:

Picture clipping. Once this function is enabled, only the pixels in the region specified by CLIP_L, CLIP_R, CLIP_T,
and CLIP_B are dumped. Please refer to **Figure 125**.
0: Disable
1: Enable

LCD:

Signaling LCD DMA. Frame ready signal is issued at the beginning of frames in Direct Couple mode, and is issued
at the end of frames in write-buffer mode. Note that in the case of direct couple mode, this function must be enabled
to trigger LCD DMA. Please refer to **Figure 124**.
0: Disable
1: Enable

DC:

Directly coupling to LCD DMA. Once this function is enabled, image data will output to LCD DMA directly instead
of writing to LCD frame buffer.
0: Disable
1: Enable

FMT:

Output frame buffer format control
00: RGB565
01: RGB888
10: ARGB8888, where A is from IBW1_ALPHA set by software.
11: Reserved

IMGDMA0+0308h Image Buffer Write 1 DMA Alpha Value Register IMGDMA0_IBW1_ALPHA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VALUE							
Type									R/W							

VALUE:

Alpha value for each pixel.

IMGDMA0+0310h Image Buffer Write 1 DMA Horizontal Size Register of Source Image IMGDMA0_IBW1_SRC_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

© 2009 MediaTek Inc.

Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the source frame. If the value is X, then it represents the source is an X-pixel wide image.

IMGDMA0+0314h Image Buffer Write 1 DMA Vertical Size Register of Source Image IMGDMA0_IBW1_SRC_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YSIZE															
Type	R/W															

YSIZE:

Vertical size of the source frame. If the value is Y, then it represents the source is a Y-line high image.

IMGDMA0+0318h Image Buffer Write 1 DMA Clip Left/Right Coordinate Register of Source Image IMGDMA0_IBW1_CLIPLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RIGHT															
Type	R/W															

LEFT:

Clipped left boundary of the source image. If the value is L, then it represents the clipping function starts from L^{th}

pixel of each line. It will be effective while IBW1_CON clip bit is enabled.

RIGHT:

Clipped right boundary of the source image. If the value is R, then it represents the clipping function finished at R^{th}

pixel of each line. It will be effective while IBW1_CON clip bit is enabled.

Please refer to **Figure 125** to get a visual understanding of the two register fields.

IMGDMA0+031Ch Image Buffer Write 1 DMA Clip Top/Bottom Coordinate Register of Source image IMGDMA0_IBW1_CLIPTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TOP															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BOTTOM															
Type	R/W															

TOP:

Clipped top boundary of the source image. If the value is T, then it represents the clipping function starts from Tth line
of the source image. It will be effective while IBW1_CON clip bit is enabled.

BOTTOM:

Clipped bottom boundary of the source image. If the value is B, then it represents the clipping function finished at Bth
line of the source image. It will be effective while IBW1_CON clip bit is enabled.

Please refer to **Figure 125** to get a visual understanding of the two register fields.

IMGDMA0+0320h Image Buffer Write 1 DMA Base Address Register of Frame Buffer 0 IMGDMA0_IBW1_BASE_ADDR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of the destination frame buffer 0. When the output format is RGB565, it should align at 2x address;
when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there
are no limits about the address setting.

IMGDMA0+0324h Image Buffer Write 1 DMA Base Address Register of Frame Buffer 1 IMGDMA0_IBW1_BASE_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;
when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there
are no limits about the address setting.
By the way, this register is only effective when IBW1 DMA is working ***under auto-restart mode***.

IMGDMA0+0328h Image Buffer Write 1 DMA Base Address Register of Frame Buffer 2 IMGDMA0_IBW1_BASE_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;

when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there

are no limits about the address setting.

By the way, this register is only effective when IBW1 DMA is working ***under auto-restart and triple-buffer mode***..

IMGDMA0+0330h Image Buffer Write 1 DMA Horizontal Size Register of Background Image 0 IMGDMA0_IBW1_BKGD_XSIZE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 0. If the value is X, then it represents the background image is an X-pixel

wide image.

The register is effective when IBW1 DMA is working ***under pitching mode***, i.e., there have existed an image in the

memory, and IBW1 DMA will replace some rectangular region of this image.

IMGDMA0+0334h Image Buffer Write 1 DMA Horizontal Size Register of Background Image 1 IMGDMA0_IBW1_BKGD_XSIZE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 1. If the value is X, then it represents the background image is an X-pixel
wide image.
The register is effective when IBW1 DMA is working ***under pitching and auto-restart mode***, i.e., there have existed
a second image in the memory, and IBW1 DMA will replace some rectangular region of the second image when the
writing base address has switched to IBW1_BASE_ADDR1.

IMGDMA0+0338h Image Buffer Write 1 DMA Horizontal Size Register of Background Image 2 IMGDMA0_IBW1_BKGD_XSIZE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 2. If the value is X, then it represents the background image is an X-pixel
wide image.
The register is effective when IBW1 DMA is working ***under pitching, auto-restart, and triple-buffer mode***, i.e.,
there have existed a third image in the memory, and IBW1 DMA will replace some rectangular region of the third
image when the writing base address has switched to IBW1_BASE_ADDR2.

IMGDMA0+0340h Image Buffer Write 1 DMA Horizontal Pixel Count Register of Received Image IMGDMA0_IBW1_RX_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal pixel count of input image. It is a 16-bit up counter. If the value is X, then it represents IBW1 DMA now
receives X^{th} pixel of a line of the input image.
Besides, while it counts to IBW1_SRC_XSIZE, it will be reset to 1.

IMGDMA0+0344h Image Buffer Write 1 DMA Vertical Line Count Register of Received Image IMGDMA0_IBW1_RX_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of input image. It is a 16-bit up counter. If the value is Y, then it represents IBW1 DMA now at Y^{th}
line of the input image.
Besides, it will increases 1 while IBW1_RX_XCNT counts to IBW1_SRC_XSIZE and IBW1 DMA receives a pixel
successfully.

Note:

When IBW1 DMA starts:
(IBW1_RX_XCNT, IBW1_RX_YCNT) = (1, 1);
When IBW1 DMA finishes:
(IBW1_RX_XCNT, IBW1_RX_YCNT) = (1, IBW1_SRC_YSIZE+1);

IMGDMA0+0350h Image Buffer Write 1 DMA Horizontal Byte Count Register of Written Image IMGDMA0_IBW1_HORI_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															COUNT	
Type															RO	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal byte count of written image. It is an 18-bit down counter. If its value is H, then it represents that there are
H bytes of a line not be written to memory yet.
Besides, while the remaining bytes could be written by a single writing command, it will be reset to IBW1_SRC_XSIZE*BPP.

IMGDMA0+0354h Image Buffer Write 1 DMA Vertical Line Count Register of Written Image IMGDMA0_IBW1_VERT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of written image. It is an 16-bit up counter. If its value is V, then it represents that IBW1 DMA now

at $(V+1)^{th}$ line of the written image.

Besides, it will increases 1 while the last write command of a line has been issued.

Note:

IBW1 DMA starts:

(IBW1_HORI_CNT, IBW1_VERT_CNT) = (IBW1_SRC_XSIZE*BPP, 0);

IBW1 DMA finishes:

(IBW1_HORI_CNT, IBW1_VERT_CNT) = (IBW1_SRC_XSIZE*BPP, IBW1_SRC_YSIZE);

IMGDMA0+0400h Image Buffer Write 2 DMA Start Register IMGDMA0_IBW2_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STR
Type																R/W
Reset																0

This register controls the activity of IBW2 DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IBW2 DMA.

0: stop IBW2 DMA

1: activate IBW2 DMA

IMGDMA0+0404h Image Buffer Write 2 DMA Control Register IMGDMA0_IBW2_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								R2Y0	IRT1	DC	CAM_CRZ_VSYNC_EN	IRT1_ULTRA_HIGH_EN	CLIP	AUTORSTR	LCD	IT
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

LCD:

Signaling LCD DMA. Frame ready signal is issued at the beginning of frames in Direct Couple mode, and is issued
at the end of frames in Dual Buffer mode. Note that in the case of automatic restart plus direct couple mode, this
function must be enabled to trigger LCD DMA.

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. IBW1 DMA automatically restarts itself while current frame is finished.

0: Disable

1: Enable

CLIP:

Picture clipping. Once this function is enabled, only the pixels in the region specified by CLIP_L, CLIP_R, CLIP_T,
and CLIP_B are dumped. Please refer to **Figure 125**.

0: Disable

1: Enable

DC:

One of multiple output options of IBW2 DMA. Directly coupling to LCD DMA. Once this function is enabled, pixel
data will output to LCD DMA directly.

0: Disable path to LCD DMA

1: Enable path to LCD DMA

IRT1:

One of multiple output options of IBW2 DMA. While the bit is set, IBW2 DMA will output pixel data to IRT1 DMA
as well as other multiple output destinations simultaneously.

0: Disable path to IRT1 DMA

1: Enable path to IRT1 DMA

R2Y0:

One of multiple output options of IBW2 DMA. While the bit is set, IBW2 DMA will output pixel data to R2Y0
(RGB2YUV module of Image Post Process module) as well as other multiple output destinations simultaneously.

0: Disable path to R2Y0

1: Enable path to R2Y0

IRT1_ULTRA_HIGH_EN:

This control bit is an **ECO solution** for enabling IRT1 ultra high function. When this port is enabled, IRT1 will get
more bandwidth from external memory.

0: Disable IRT1 ultra high function

1: Enable IRT1 ultra high function

CAM_CRZ_VSYNC_EN:

This control bit is an **ECO solution** for controlling the frame synchronous of OVL DMA and OVL_MO_0 to camera and CRZ. Notice it should be configured only when OVL_MO_1 output to VDOENC WDMA (video preview / video capture scenario) .

0: Disable frame synchronous to camera and CRZ

1: Disable frame synchronous to camera and CRZ

IMGDMA0+0408h Image Buffer Write 2 DMA Alpha Value Register IMGDMA0_IBW2_ALPHA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**VALUE**							
Type									R/W							

VALUE:

Alpha value for each pixel.

Because the input pixels are 24-bits RGB data, but data interface between IBW2 DMA to LCD / IBW2 DMA to IRT1

DMA are 32-bit wide, therefore it will prefix this alpha value to construct a ARGB pixel.

IMGDMA0+0410h Image Buffer Write 2 DMA Horizontal Size Register of Source Image IMGDMA0_IBW2_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**XSIZE**															
Type	R/W															

XSIZE:

Horizontal size of the source frame. If the value is X, then it represents the source is an X-pixel wide image.

IMGDMA0+0414h Image Buffer Write 2 DMA Vertical Size Register of Source Image IMGDMA0_IBW2_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**YSIZE**															
Type	R/W															

YSIZE:

Vertical size of the source frame. If the value is Y, then it represents the source is a Y-line high image.

 © 2009 MediaTek Inc.

IMGDMA0+0418h Image Buffer Write 2 DMA Clip Left/Right Coordinate Register of Source Image IMGDMA0_IBW2_CLIPLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LEFT															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RIGHT															
Type	R/W															

LEFT:

Clipped left boundary of the source image. If the value is L, then it represents the clipping function starts from L^{th} pixel of each line. It will be effective while IBW2_CON clip bit is enabled.

RIGHT:

Clipped right boundary of the source image. If the value is R, then it represents the clipping function finished at R^{th} pixel of each line. It will be effective while IBW2_CON clip bit is enabled.

Please refer to **Figure 125** to get a visual understanding of the two register fields.

IMGDMA0+041Ch Image Buffer Write 2 DMA Clip Top/Bottom Coordinate Register of Source image IMGDMA0_IBW2_CLIPTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TOP															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BOTTOM															
Type	R/W															

TOP:

Clipped top boundary of the source image. If the value is T, then it represents the clipping function starts from T^{th} line of the source image. It will be effective while IBW2_CON clip bit is enabled.

BOTTOM:

Clipped bottom boundary of the source image. If the value is B, then it represents the clipping function finished at B^{th} line of the source image. It will be effective while IBW2_CON clip bit is enabled.

Please refer to **Figure 125** to get a visual understanding of the two register fields.

IMGDMA0+0420h Image Buffer Write 2 DMA Horizontal Pixel Count Register of Received Image IMGDMA0_IBW2_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal pixel count of input image. It is a 16-bit up counter. If the value is X, then it represents IBW2 DMA now
receives X^{th} pixel of a line of the input image.
Besides, while it counts to IBW2_XSIZE, it will be reset to 1.

IMGDMA0+0424h Image Buffer Write 2 DMA Vertical Line Count Register of Received Image IMGDMA0_IBW2_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of input image. It is a 16-bit up counter. If the value is Y, then it represents IBW2 DMA now at Y^{th}
line of the input image.
Besides, it will increases 1 while IBW2_XCNT counts to IBW2_XSIZE and IBW2 DMA receives a pixel
successfully.

Note:

When IBW2 DMA starts:
(IBW2_XCNT, IBW2_YCNT) = (1, 1);
When IBW2 DMA finishes:
(IBW2_XCNT, IBW2_YCNT) = (1, IBW2_YSIZE+1);

IMGDMA0+0500h Image Buffer Read 1 DMA Start Register IMGDMA0_IBR1_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STR
Type																R/W
Reset																0

This register controls the activity of IBR1 DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IBR1 DMA.
0: stop IBR1 DMA
1: activate IBR1 DMA

IMGDMA0+0504h Image Buffer Read 1 DMA Control Register IMGDMA0_IBR1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**ORDER**	**FMT**	**IT**
Type														R/W	R/W	R/W
Reset														0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

FMT:

Data format of frame buffer.

0: The data is two bytes per pixel, i.e., the data format of frame buffer is RGB565.

1: The data is three bytes per pixel, i.e., the data format of frame buffer is either RGB888 or BGR888.

ORDER:

Data order of a 3-byte pixel data. It determines the data format is RGB888 or BGR888.

0: BGR888, from MSB to LSB (R is at lower memory location).

1: RGB888, from MSB to LSB (B is at lower memory location).

Please refer to **Figure 127** to realize the byte sequence determined by FMT and ORDER.

IMGDMA0+0508h Image Buffer Read 1 DMA Base Address Register of Source Image IMGDMA0_IBR1_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of the source image buffer. ***It should align at 8x address***, or IBR1 DMA will read wrong data.

IMGDMA0+050Ch Image Buffer Read 1 DMA Pixel Number Register of Source Image IMGDMA0_IBR1_PXLNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**NUM**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**NUM**															
Type	R/W															

NUM:

Number of pixels of the source image. If its value is N, then it represents the source is an N-pixel image.

IMGDMA0+0510h Image Buffer Read 1 DMA Pixel Count Register of Source Image IMGDMA0_IBR1_PXLCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COUNT															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Pixel Counter. It is a up counter. If its value is C, then it represents IBR1 now transfers (C+1) th pixel to R2Y0

module.

Note:

When IBR1 starts:

IBR1_PXLCNT = 0;

When IBR1 finishes:

IBR1_PXLCNT = IBR1_PXLNUM;

IMGDMA0+0700h Overlay DMA Start Register IMGDMA0_OVL_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STR
Type																R/W
Reset																0

This register controls the activity of OVL DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for OVL DMA.

0: stop OVL DMA

1: activate OVL DMA

Note:

OVL DMA needs to be set only when the overlay function is enabled. If OVL DMA just bypass the pixel data, the

only register needs to be set is ‘psel’ bit of OVL_CON to select pixel data source, either from IPP1 (Image Post

Processor) or CRZ (Capture Resizer).

IMGDMA0+0704h **Overlay DMA Control Register** **IMGDMA0_OVL_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PSEL	PALEN		MODE 1	MODE 0	IT
Type											R/W	R/W		R/W	R/W	R/W
Reset											0	0		0	0	0

IT:

Interrupt Enabling. It is valid only when OVL DMA is started.

0: Disable

1: Enable

MODE:

Mask data format. OVL DMA will read mask data from memory, and then overlay the input pixel data according the

mask data. When value of mask data is equal to color key of OVL_CFG, OVL DMA will bypass input pixel data to

output; else, a new color is used to replace the current input pixel to output.

00: 1-bit per pixel of mask data

01: 2-bit per pixel of mask data

10: 4-bit per pixel of mask data

11: 8-bit per pixel of mask data

PALEN:

Photo frame palette enabling. Please set this bit before any operation with the palette memory. The palette memory is

read/write through APB bus.

0: Palette read/write disable

1: Palette read/write enable

PSEL:

Pixel engine selection.

0: IPP1

1: CRZ

Note:

There exists a memory power off control bit (AP_GRAPH1_MEM_PWR_OFF bit 7) to turn off the OVL DMA palette. Before using OVL DMA palette, SW should make sure that the control bit is set to 0.

IMGDMA0+0708h Overlay DMA Base Address Register of Mask Image IMGDMA0_OVL_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the source image buffer. ***It should align at 8x address***, or OVL DMA will read wrong data.

IMGDMA0+070Ch Overlay DMA Configuration Register IMGDMA0_OVL_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COLOR KEY								VRATIO				HRATIO			
Type	R/W								R/W				R/W			

COLOR KEY:

Transparent color key for overlay function. If value of mask data equal to the color key, then OVL DMA will bypass input pixel data to output.

VRATIO:

Vertical scaling ratio. It is used to scaling the vertical size of mask image to the vertical size of source image received from the pixel engine. Due to VRATIO is 4-bits, the scaling ratio could only be 1 to 15. If the value is V, then OVL DMA will scale vertical size of make image V-ple.

HRATIO:

Horizontal scaling ratio. It is used to scaling the horizontal size of mask image to the horizontal size of source image received from the pixel engine. Due to HRATIO is 4-bits, the scaling ratio could only be 1 to 15. If the value is H, then OVL DMA will scale horizontal size of make image H-ple.

IMGDMA0+0710h Overlay DMA Horizontal Size Register of Mask Image IMGDMA0_OVL_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															

Type	R/W

XSIZE:

Horizontal size of the make image. If the value is X, then it represents the mask is an X-pixel wide image.

IMGDMA0+0714h Overlay DMA Vertical Size Register of Mask Image IMGDMA0_OVL_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YSIZE															
Type	R/W															

YSIZE:

Vertical size of the mask image. If the value is Y, then it represents the mask is a Y-line high image.

IMGDMA0+0718h Overlay DMA Horizontal Pixel Count Register of Mask Image IMGDMA0_OVL_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal pixel count of the make image. It is a 16-bit up counter. If the value is X, then it represents OVL DMA

now transmits $(X+1)^{th}$ pixel of a line of the make image.

Besides, while it counts to (OVL_XSIZE – 1), it will be reset to 0. And this counter is effective only when overlay

function is enabled.

IMGDMA0+071Ch OVL DMA Vertical Line Count Register of Mask Image IMGDMA0_OVL_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of the mask image. It is a 16-bit up counter. If the value is Y, then it represents OVL DMA now at

$(Y+1)^{th}$ line of the make image.

© 2009 MediaTek Inc.

Besides, it will increases 1 while OVL_XCNT counts to (OVL_XSIZE – 1). And if OVL_YCNT counts to
(OVL_YSIZE – 1), it will be reset to 0.

Note:

When OVL DMA starts or finishes:

(OVL_XCNT, OVL_YCNT) = (0, 0);

IMGDMA0+0800h OVL DMA Palette Register 00 IMGDMA0_OVL_PAL_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									COLOR Y							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COLOR U								COLOR V							
Type	R/W															

This register stores the colors of the first entry of the palette. Note that the second entry is at IMGDMA0+0804h, the third entry is at IMGDMA0+0808h, and so on. The last entry (256th) is at IMGDMA0+0xBFCh.

COLOR Y:

Palette entry Y color value.

COLOR U:

Palette entry U color value.

COLOR V:

Palette entry V color value.

Note:

Before read/write the palette memory, remember to set palen bit of OVL_CON first.

IMGDMA0+0780h Overlay DMA Multiple Output Engine 0 Start Register IMGDMA0_OVL_MO_0_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STR
Type																R/W
Reset																0

This register controls the activity of OVL Multiple Output 0. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for OVL Multiple Output 0

0: stop OVL Multiple Output 0

1: activate OVL Multiple Output 0

Note:

OVL Multiple Output module is used to transmit OVL DMA output pixel data to multiple destinations simultaneously, including JPEG DMA, VDOENC WDMA, PRZ, DRZ, and Y2R0. Due to its destination located at

Graphsys1 and Graphsys2 respectively, in order to transmit pixel data to different destinations located at different

Graphsys, OVL Multiple Output has been divided into OVL Multiple Output 0 and OVL Multiple Output 1 to support

multiple output function. ***The starting sequence of OVL DMA, OVL Multiple Output 0 and OVL Multiple Output 1***

should be as followed.

OVL Multiple Output 1 started → OVL Multiple Output 0 started → OVL DMA

If OVL DMA is allowed to transmit pixel data before OVL Multiple Output 0/1 are started, it would lose some pixels.

IMGDMA0+0784h Overlay DMA Multiple Output Engine 0 Control Register IMGDMA0_OVL_MO_0_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**DRZ**	**Y2R0**	**IMGDMA1**
Type														R/W	R/W	R/W
Reset														0	0	0

IMGDMA1:

Enable output to OVL Multiple Output 1 in Image DMA 1. Then OVL Multiple Output 1 will output data to

VDOENC WDMA, JPEG DMA, or PRZ.

0: Disable

1: Enable

Y2R0:

Enable output to Y2R0.

0: Disable

1: Enable

DRZ:

Enable output to DRZ.

0: Disable

1: Enable

IMGDMA0+0788h Overlay DMA Multiple Output Engine 0 Busy Status IMGDMA0_OVL_MO_0_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**DRZ BUSY**	**Y2R0 BUSY**	**IMGDMA1 BUSY**
Type														RO	RO	RO

The BUSY bits are used to show if OVL Multiple Output 0 transmits data to destination or not. If there are two or more output path are enabled and only some BUSY bit stays at 1, then the OVL Multiple Output 0 may be stalled by the path.

IMGDMA1 BUSY:

0: No data transmit to Image DMA 1 now.

1: Data are transmitted to Image DMA 1 now.

Y2R0 BUSY:

0: No data transmit to Y2R0 now.

1: Data are transmitted to Y2R0 now.

DRZ BUSY:

0: No data transmit to DRZ now.

1: Data are transmitted to DRZ now.

IMGDMA0+0C80h Image Rotator 0 DMA Multiple Output Engine 0 Start Register IMGDMA0_IRT0_MO_0_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IRT0 Multiple Output 0. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IRT0 Multiple Output 0

0: stop IRT0 Multiple Output 0

1: activate IRT0 Multiple Output 0

Note:

IRT0 Multiple Output module is used to transmit output pixel of either IRT0 DMA or VDODEC DMA with

Scan-Line mode to multiple destinations simultaneously, including MP4DEBLK, CRZ, PRZ, and IPP1. Due to its

destination located at Graphsys1 and Graphsys2 respectively, in order to transmit pixel data to different destinations

 © 2009 MediaTek Inc.

located at different Graphsys, IRT0 Multiple Output has been divided into IRT0 Multiple Output 0 and IRT0 Multiple
Output 1 to support multiple output function. ***The starting sequence of IRT0 DMA/VDODEC DMA with Scan-Line***
mode, IRT0 Multiple Output 0 and IRT0 Multiple Output 1 should be as followed.
IRT0 Multiple Output 0 started → IRT0 Multiple Output 1 started → IRT0 DMA / VDODEC DMA with
Scan-Line Mode
If IRT0 DMA / VDODEC DMA with Scan-Line Mode is allowed to transmit pixel data before IRT0 Multiple Output
0/1 are started, it would lose some pixels.

IMGDMA0+0C84h Image Rotator 0 DMA Multiple Output Engine 0 Control Register IMGDMA0_IRT0_MO_0_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**IPP1**	**CRZ**
Type															R/W	R/W
Reset															0	0

CRZ:

Enable output to CRZ.

0: Disable

1: Enable

IPP1:

Enable output to IPP1.

0: Disable

1: Enable

IMGDMA0+0C88h Image Rotator 0 DMA Multiple Output Engine 0 Busy Status IMGDMA0_IRT0_MO_0_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**IPP1 BUSY**	**CRZ BUSY**
Type															RO	RO

The BUSY bits are used to show if IRT0 Multiple Output 0 transmits data to destination or not. If there are two or more output path are enabled and only some BUSY bit stays at 1, then the IRT0 Multiple Output 0 may be stalled by the path.

CRZ BUSY:

0: No data transmit to CRZ now.

1: Data are transmitted to CRZ now.

IPP1 BUSY:

0: No data transmit to IPP1 now.

1: Data are transmitted to IPP1 now.

IMGDMA0+0D00h Image Rotator 1 DMA Start Register IMGDMA0_IRT1_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IRT1 DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IRT1 DMA.

0: stop IRT1 DMA

1: activate IRT1 DMA

IMGDMA0+0D04h Image Rotator 1 DMA Control Register IMGDMA0_IRT1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**BUFFER DOUBLE**	**FMT**		**FLIP**	**ROT**		**PITCH**	**LCD**	**TRIPLE**	**AUTO RSTR**	**IT**
Type						R/W	R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset						0	0		0	0		0	0	0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. IRT1 DMA automatically restarts itself while current frame is finished.

0: Disable

1: Enable

TRIPLE:

When automatic restart function is enable, it could enable this bit to support triple frame buffers; else, two frame
buffers are supported under auto-restart mode.
0: Two Frame Buffer Supported
1: Three Frame Buffer Supported

LCD:

Signaling LCD DMA. Frame ready signal is issued at the end of writing data to frame buffer. Please refer to section
Image Rotator 1 / Image Rotator 3 DMA for more details.
0: Disable
1: Enable

PITCH:

Destination pitching. Please refer to **Figure 126** to get a quick understanding.
0: Disable
1: Enable

ROT:

Rotation direction related to input image. Please refer to **Figure 130** to get the definition of rotation.
00: No rotation
01: 90° rotation
10: 180° rotation
11: 270° rotation

FLIP:

Flipping option related to the rotated image. Please refer to **Figure 130** to get the definition of flipping.
0: No flip
1: Flipped after rotation

FMT:

Output frame buffer format control
00: RGB565
01: RGB888
10: ARGB8888, where A is from IBW2_DMA.
11: ARGB8888, where A is from IRT1_ALPHA set by software.

BUFFER DOUBLE:

When rotation direction is 90° or 270° , it needs to allocate a piece of memory to serve as line buffer to achieve the
two kinds of rotations. If one of the two rotations is enabled, it needs at least IRT1_SRC_XSIZE * 32 bytes to serve
as line buffer. And if the BUFFER DOUBLE bit is enabled, it will use IRT1_SRC_XSIZE * 64 bytes to serve as line
buffer.
0: Disable
1: Enable

 © 2009 MediaTek Inc.

IMGDMA0+0D08h Image Rotator 1 DMA Alpha Value Register IMGDMA0_IRT1_ALPHA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**VALUE**							
Type									R/W							

VALUE:

Alpha value for each pixel.

IMGDMA0+0D10h Image Rotator 1 DMA Horizontal Size Register of Source Image IMGDMA0_IRT1_SRC_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**XSIZE**															
Type	R/W															

XSIZE:

Horizontal size of the source frame. If the value is X, then it represents the source is an X-pixel wide image.

IMGDMA0+0D14h Image Rotator 1 DMA Vertical Size Register of Source Image IMGDMA0_IRT1_SRC_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**YSIZE**															
Type	R/W															

YSIZE:

Vertical size of the source frame. If the value is Y, then it represents the source is a Y-line high image.

IMGDMA0+0D20h Image Rotator 1 DMA Base Address Register of Frame Buffer 0 IMGDMA0_IRT1_BASE_ADDR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of the destination frame buffer 0. When the output format is RGB565, it should align at 2x address;

when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there
are no limits about the address setting.

IMGDMA0+0D24h Image Rotator 1 DMA Base Address Register of Frame Buffer 1 IMGDMA0_IRT1_BASE_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;
when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there
are no limits about the address setting.
By the way, this register is only effective when IRT1 DMA is working ***under auto-restart mode***.

IMGDMA0+0D28h Image Rotator 1 DMA Base Address Register of Frame Buffer 2 IMGDMA0_IRT1_BASE_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;
when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there
are no limits about the address setting.
By the way, this register is only effective when IRT1 DMA is working ***under auto-restart and triple-buffer mode***..

IMGDMA0+0D30h Image Rotator 1 DMA Horizontal Size Register of Background Image 0 IMGDMA0_IRT1_BKGD_XSIZE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

© 2009 MediaTek Inc.

XSIZE:

Horizontal size of the background image 0. If the value is X, then it represents the background image is an X-pixel
wide image.

The register is effective when IRT1 DMA is working ***under pitching mode***, i.e., there have existed an image in the
memory, and IRT1 DMA will replace some rectangular region of this image.

IMGDMA0+0D34h Image Rotator 1 DMA Horizontal Size Register of Background Image 1 IMGDMA0_IRT1_BKGD_XSIZE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 1. If the value is X, then it represents the background image is an X-pixel
wide image.

The register is effective when IRT1 DMA is working ***under pitching and auto-restart mode***, i.e., there have existed
a second image in the memory, and IRT1 DMA will replace some rectangular region of the second image when the
writing base address has switched to IRT1_BASE_ADDR1.

IMGDMA0+0D38h Image Rotator 1 DMA Horizontal Size Register of Background Image 2 IMGDMA0_IRT1_BKGD_XSIZE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 2. If the value is X, then it represents the background image is an X-pixel
wide image.

The register is effective when IRT1 DMA is working ***under pitching, auto-restart, and triple-buffer mode***, i.e.,
there have existed a third image in the memory, and IRT1 DMA will replace some rectangular region of the third
image when the writing base address has switched to IRT1_BASE_ADDR2.

IMGDMA0+0D40h Image Rotator 1 DMA Base Address Register of Line Buffer IMGDMA0_IRT1_FIFO_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of line buffer which is used when rotation is 90° or 270° . ***It should align at 8x address***, or IRT1

DMA will read wrong data. Besides, it should be set in the internal memory to get better performance.

Note 1:

Notice that IRT1 DMA is inside graph1sys, so the internal memory address would be better ranging from

0x4000_0000 to 0x4001_7fff than from 0x4002_0000 to 0x4004_3fff.

Note 2:

When rotation is 90° or 270° , IRT1 DMA will use either (IRT1_SRC_XSIZE * 32) bytes or (IRT1_SRC_XSIZE *

64) bytes to serve as line buffer according to the setting of BUFFER DOUBLE bit of IRT1_CON.

IMGDMA0+0D50h Image Rotator 1 DMA Horizontal Pixel Count Register of Received Image IMGDMA0_IRT1_RX_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

Horizontal pixel count of input image. It is a 16-bit up counter. If the value is X, then it represents IRT1 DMA now

receives X^{th} pixel of a line of the input image.

Besides, while it counts to IRT1_SRC_XSIZE, it will be reset to 1.

IMGDMA0+0D54h Image Rotator 1 DMA Vertical Line Count Register of Received Image IMGDMA0_IRT1_RX_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

Vertical line count of input image. It is a 16-bit up counter. If the value is Y, then it represents IRT1 DMA now at Y[th]
line of the input image.
Besides, it will increases 1 while IRT1_RX_XCNT counts to IRT1_SRC_XSIZE and IRT1 DMA receives a pixel
successfully.

Note:
When IRT1 DMA starts:
(IRT1_RX_XCNT, IRT1_RX_YCNT) = (1, 1);
When IRT1 DMA finishes:
(IRT1_RX_XCNT, IRT1_RX_YCNT) = (1, IRT1_SRC_YSIZE+1);

IMGDMA0+0D60h Image Rotator 1 DMA Horizontal Byte Count Register of Written Image IMGDMA0_IRT1_HORI_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														COUNT		
Type														RO		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal byte count of written image. It is an ***signed*** 19-bit down counter. It works at two modes.
When rotation is 90° or 270° :
It could not tell how many bytes of a line are not written yet. It only tell if a line is finished or not. At
IRT1 DMA starts, its value will be IRT1_SRC_XSIZE*BPP ; at the end of a frame, it should be zero or
negative.
When rotation is 0° or 180° :
If its value is H, then it represents that there are H bytes of a line not be written to memory yet.
Besides, while the remaining bytes could be written by a single writing command, it will be reset to
IRT1_SRC_XSIZE*BPP.

IMGDMA0+0D64h Image Rotation 1 DMA Vertical Line Count Register of Written Image IMGDMA0_IRT1_VERT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of written image. It is an 16-bit up counter. If its value is V, then it represents that IRT1 DMA now

at $(V+1)^{th}$ line of the written image. It also works at two modes.

When rotation is 90° or 270° :

It increases 1 while IRT1 DMA issue a write command to process the data of current segment. Sometimes

there are no data needed to be processed, and the counter just increases 1 and no action is performed. And

it will be reset to 0 while it counts to (IRT1_SRC_YSIZE – 1) and an increase signal is issued.

When rotation is 0° or 180° :

It will increases 1 while the last write command of a line has been issued.

Note:

IRT1 DMA starts:

(IRT1_HORI_CNT, IRT1_VERT_CNT) = (IRT1_SRC_XSIZE*BPP, 0);

IRT1 DMA finishes:

When rotation is 90° or 270° :

(IRT1_HORI_CNT, IRT1_VERT_CNT) = (0 or -1, 0);

When rotation is 0° or 180° :

(IRT1_HORI_CNT, IRT1_VERT_CNT) = (IRT1_SRC_XSIZE*BPP, IRT1_SRC_YSIZE);

IMGDMA0+0F00h Image Rotator 3 DMA Start Register IMGDMA0_IRT3_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IRT3 DMA. Note that before setting STR to "1", all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IRT3 DMA.

0: stop IRT3 DMA

1: activate IRT3 DMA

IMGDMA0+0F04h Image Rotator 3 DMA Control Register IMGDMA0_IRT3_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**BUFFER DOUBLE**	**FMT**		**FLIP**	**ROT**		**PITCH**		**TRIPLE**	**AUTO RSTR**	**IT**
Type						R/W	R/W		R/W	R/W		R/W		R/W	R/W	R/W
Reset						0	0		0	0		0		0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. IRT3 DMA automatically restarts itself while current frame is finished.

0: Disable

1: Enable

TRIPLE:

When automatic restart function is enable, it could enable this bit to support triple frame buffers; else, two frame buffers are supported under auto-restart mode.

0: Two Frame Buffer Supported

1: Three Frame Buffer Supported

PITCH:

Destination pitching. Please refer to **Figure 126** to get a quick understanding.

0: Disable

1: Enable

ROT:

Rotation direction related to input image. Please refer to **Figure 130** to get the definition of rotation.

00: No rotation

01: 90° rotation

10: 180° rotation

11: 270° rotation

FLIP:

Flipping option related to the rotated image. Please refer to **Figure 130** to get the definition of flipping.

0: No flip

1: Flipped after rotation

FMT:

Output frame buffer format control

00: RGB565

01: RGB888

10: ARGB8888, where A is from LCD.

11: ARGB8888, where A is from IRT3_ALPHA set by software.

BUFFER DOUBLE:

When rotation direction is 90° or 270°, it needs to allocate a piece of memory to serve as line buffer to achieve the
two kinds of rotations. If one of the two rotations is enabled, it needs at least IRT3_SRC_XSIZE * 32 bytes to serve
as line buffer. And if the BUFFER DOUBLE bit is enabled, it will use IRT3_SRC_XSIZE * 64 bytes to serve as line
buffer.

0: Disable

1: Enable

IMGDMA0+0F08h Image Rotator 3 DMA Alpha Value Register IMGDMA0_IRT3_ALPHA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VALUE							
Type									R/W							

VALUE:

Alpha value for each pixel.

IMGDMA0+0F10h Image Rotator 3 DMA Horizontal Size Register of Source Image IMGDMA0_IRT3_SRC_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the source frame. If the value is X, then it represents the source is an X-pixel wide image.

IMGDMA0+0F14h Image Rotator 3 DMA Vertical Size Register of Source Image IMGDMA0_IRT3_SRC_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YSIZE															
Type	R/W															

YSIZE:

Vertical size of the source frame. If the value is Y, then it represents the source is a Y-line high image.

 © 2009 MediaTek Inc.

IMGDMA0+0F20h Image Rotator 3 DMA Base Address Register of Frame Buffer 0 IMGDMA0_IRT3_BASE_ADDR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 0. When the output format is RGB565, it should align at 2x address;

when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there

are no limits about the address setting.

IMGDMA0+0F24h Image Rotator 3 DMA Base Address Register of Frame Buffer 1 IMGDMA0_IRT3_BASE_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;

when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there

are no limits about the address setting.

By the way, this register is only effective when IRT3 DMA is working ***under auto-restart mode***.

IMGDMA0+0F28h Image Rotator 3 DMA Base Address Register of Frame Buffer 2 IMGDMA0_IRT3_BASE_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the destination frame buffer 1. When the output format is RGB565, it should align at 2x address;

when the output format is ARGB8888, it should align at 4x address; else, when the output format is RGB888, there

© 2009 MediaTek Inc.

are no limits about the address setting.

By the way, this register is only effective when IRT3 DMA is working ***under auto-restart and triple-buffer mode***..

IMGDMA0+0F30h Image Rotator 3 DMA Horizontal Size Register of Background Image 0 IMGDMA0_IRT3_BKGD_XSIZE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 0. If the value is X, then it represents the background image is an X-pixel

wide image.

The register is effective when IRT3 DMA is working ***under pitching mode***, i.e., there have existed an image in the

memory, and IRT3 DMA will replace some rectangular region of this image.

IMGDMA0+0F34h Image Rotator 3 DMA Horizontal Size Register of Background Image 1 IMGDMA0_IRT3_BKGD_XSIZE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															
Type	R/W															

XSIZE:

Horizontal size of the background image 1. If the value is X, then it represents the background image is an X-pixel

wide image.

The register is effective when IRT3 DMA is working ***under pitching and auto-restart mode***, i.e., there have existed

a second image in the memory, and IRT3 DMA will replace some rectangular region of the second image when the

writing base address has switched to IRT3_BASE_ADDR1.

IMGDMA0+0F38h Image Rotator 3 DMA Horizontal Size Register of Background Image 2 IMGDMA0_IRT3_BKGD_XSIZE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XSIZE															

 © 2009 MediaTek Inc.

Type	R/W

XSIZE:

Horizontal size of the background image 2. If the value is X, then it represents the background image is an X-pixel
wide image.
The register is effective when IRT3 DMA is working ***under pitching, auto-restart, and triple-buffer mode***, i.e.,
there have existed a third image in the memory, and IRT3 DMA will replace some rectangular region of the third
image when the writing base address has switched to IRT3_BASE_ADDR2.

IMGDMA0+0F40h Image Rotator 3 DMA Base Address Register of Line Buffer IMGDMA0_IRT3_FIFO_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of line buffer which is used when rotation is 90° or 270° . ***It should align at 8x address***, or IRT3
DMA will read wrong data. Besides, it should be set in the internal memory to get better performance.
Note 1:
Notice that IRT3 DMA is inside graph1sys, so the internal memory address would be better ranging from
0x4000_0000 to 0x4001_7fff than from 0x4002_0000 to 0x4004_3fff.
Note 2:
When rotation is 90° or 270° , IRT3 DMA will use either (IRT3_SRC_XSIZE * 32) bytes or (IRT3_SRC_XSIZE *
64) bytes to serve as line buffer according to the setting of BUFFER DOUBLE bit of IRT3_CON.

IMGDMA0+0F50h Image Rotator 3 DMA Horizontal Pixel Count Register of Received Image IMGDMA0_IRT3_RX_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

Horizontal pixel count of input image. It is a 16-bit up counter. If the value is X, then it represents IRT3 DMA now
receives X^{th} pixel of a line of the input image.

Besides, while it counts to IRT3_SRC_XSIZE, it will be reset to 1.

IMGDMA0+0F54h Image Rotator 3 DMA Vertical Line Count Register of Received Image IMGDMA0_IRT3_RX_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

Vertical line count of input image. It is a 16-bit up counter. If the value is Y, then it represents IRT3 DMA now at Y^{th}
line of the input image.
Besides, it will increases 1 while IRT3_RX_XCNT counts to IRT3_SRC_XSIZE and IRT3 DMA receives a pixel
successfully.

Note:
When IRT3 DMA starts:
(IRT3_RX_XCNT, IRT3_RX_YCNT) = (1, 1);
When IRT3 DMA finishes:
(IRT3_RX_XCNT, IRT3_RX_YCNT) = (1, IRT3_SRC_YSIZE+1);

IMGDMA0+0F60h Image Rotator 3 DMA Horizontal Byte Count Register of Written Image IMGDMA0_IRT3_HORI_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														**COUNT**		
Type														RO		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

Horizontal byte count of written image. It is an ***signed*** 18-bit down counter. It works at two modes.
When rotation is 90° or 270° :
It could not tell how many bytes of a line are not written yet. It only tell if a line is finished or not. At
IRT3 DMA starts, its value will be IRT3_SRC_XSIZE*BPP ; at the end of a frame, it should be zero or
negative.
When rotation is 0° or 180° :
If its value is H, then it represents that there are H bytes of a line not be written to memory yet.

Besides, while the remaining bytes could be written by a single writing command, it will be reset to

IRT3_SRC_XSIZE*BPP.

IMGDMA0+0F64h Image Rotation 3 DMA Vertical Line Count Register of Written Image IMGDMA0_IRT3_VERT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of written image. It is an 16-bit up counter. If its value is V, then it represents that IRT3 DMA now at (V+1)th line of the written image. It also works at two modes.

When rotation is 90° or 270° :

It increases 1 while IRT3 DMA issue a write command to process the data of current segment. Sometimes there are no data needed to be processed, and the counter just increases 1 and no action is performed. And it will be reset to 0 while it counts to (IRT3_SRC_YSIZE – 1) and an increase signal is issued.

When rotation is 0° or 180° :

It will increases 1 while the last write command of a line has been issued.

Note:

IRT3 DMA starts:

(IRT3_HORI_CNT, IRT3_VERT_CNT) = (IRT3_SRC_XSIZE*BPP, 0);

IRT3 DMA finishes:

When rotation is 90° or 270° :

(IRT3_HORI_CNT, IRT3_VERT_CNT) = (0 or -1, 0);

When rotation is 0° or 180° :

(IRT3_HORI_CNT, IRT3_VERT_CNT) = (IRT3_SRC_XSIZE*BPP, IRT3_SRC_YSIZE);

5.16.2.2 Image DMA 1

Register Address	Register Function	Acronym
IMGDMA1+0000h	Image DMA 1 Status Register	**IMGDMA1_STA**
IMGDMA1+0004h	Image DMA 1 Interrupt Acknowledge Register	**IMGDMA1_ACKINT**
IMGDMA1+0020h	Image DMA 1 GMC Interface Status	**IMGDMA1_GMCIF_STA**
IMGDMA1+0100h	JPEG DMA Start Register	**IMGDMA1_JPEG_STR**

IMGDMA1+0104h	JPEG DMA Control Register	**IMGDMA1_JPEG_CON**
IMGDMA1+0110h	JPEG DMA Base Address Register of Line FIFO	**IMGDMA1_JPEG_FIFO_BASE**
IMGDMA1+0114h	JPEG DMA FIFO Length Register	**IMGDMA1_JPEG_FIFOLEN**
IMGDMA1+0120h	JPEG DMA Horizontal Size Register of Source Image	**IMGDMA1_JPEG_XSIZE**
IMGDMA1+0124h	JPEG DMA Vertical Size Register of Source Image	**IMGDMA1_JPEG_YSIZE**
IMGDMA1+0130h	JPEG DMA Write Pointer Register	**IMGDMA1_JPEG_WRPTR**
IMGDMA1+0134h	JPEG DMA Write Horizontal Pixel Count Register of Received Image	**IMGDMA1_JPEG_WRXCNT**
IMGDMA1+0138h	JPEG DMA Write Vertical Line Count Register of Received Image	**IMGDMA1_JPEG_WRYCNT**
IMGDMA1+0140h	JPEG DMA Read Pointer Register	**IMGDMA1_JPEG_RDPTR**
IMGDMA1+0144h	JPEG DMA Read Horizontal Block Count Register of Padded Image	**IMGDMA1_JPEG_RDXBLK_CNT**
IMGDMA1+0148h	JPEG DMA Read Vertical Block Count Register of Padded Image	**IMGDMA1_JPEG_RDYBLK_CNT**
IMGDMA1+0150h	JPEG DMA FIFO Line Count Register	**IMGDMA1_JPEG_FFCNT**
IMGDMA1+0154h	JPEG DMA FIFO Write Line Index Register	**IMGDMA1_JPEG_FFWRLIDX**
IMGDMA1+0158h	JPEG DMA FIFO Read Line Index Register	**IMGDMA1_JPEG_FFRDLIDX**
IMGDMA1+0200h	Video Encode DMA Start Register	**IMGDMA1_VDOENC_STR**
IMGDMA1+0204h	Video Encode DMA Control Register	**IMGDMA1_VDOENC_CON**
IMGDMA1+0210h	Video Encode DMA Y Base Address 1 Register	**IMGDMA1_VDOENC_Y_BASE1**
IMGDMA1+0214h	Video Encode DMA U Base Address 1 Register	**IMGDMA1_VDOENC_U_BASE1**
IMGDMA1+0218h	Video Encode DMA V Base Address 1 Register	**IMGDMA1_VDOENC_V_BASE1**
IMGDMA1+0220h	Video Encode DMA Y Base Address 2 Register	**IMGDMA1_VDOENC_Y_BASE2**
IMGDMA1+0224h	Video Encode DMA U Base Address 2 Register	**IMGDMA1_VDOENC_U_BASE2**

© 2009 MediaTek Inc.

IMGDMA1+0228h	Video Encode DMA V Base Address 2 Register	**IMGDMA1_VDOENC_V_BASE2**
IMGDMA1+0230h	Video Encode DMA Horizontal Size Register of Source Image	**IMGDMA1_VDOENC_XSIZE**
IMGDMA1+0234h	Video Encode DMA Vertical Size Register of Source Image	**IMGDMA1_VDOENC_YSIZE**
IMGDMA1+0238h	Video Encode DMA Pixel Number Register of Source Image	**IMGDMA1_VDOENC_PXLNUM**
IMGDMA1+0240h	Video Encode Write DMA Horizontal Pixel Count Register of Written Image	**IMGDMA1_VDOENC_WXCNT**
IMGDMA1+0244h	Video Encode Write DMA Vertical Line Count Register of Written Image	**IMGDMA1_VDOENC_WYCNT**
IMGDMA1+0250h	Video Encode Read DMA Y component Horizontal Pixel Count Register	**IMGDMA1_VDOENC_Y_XCNT**
IMGDMA1+0254h	Video Encode Read DMA Y component Vertical Line Count Register	**IMGDMA1_VDOENC_Y_YCNT**
IMGDMA1+0258h	Video Encode Read DMA V component Horizontal Pixel Count Register	**IMGDMA1_VDOENC_V_XCNT**
IMGDMA1+025Ch	Video Encode Read DMA V component Vertical Line Count Register	**IMGDMA1_VDOENC_V_YCNT**
IMGDMA1+0260h	Video Encode Read DMA Dropped Frame Count Register	**IMGDMA1_VDOENC_DROP_FCNT**
IMGDMA1+0270h	Video Encode DMA Line Buffer Base Address Register	**IMGDMA1_VDOENC_LB_BASE**
IMGDMA1+0274h	Video Encode DMA Line Buffer Length Register	**IMGDMA1_VDOENC_LB_YLEN**
IMGDMA1+0278h	Video Encode DMA Line Buffer Write Engine Horizontal Pixel Count Register	**IMGDMA1_VDOENC_LB_WR_XCNT**
IMGDMA1+027Ch	Video Encode DMA Line Buffer Write Engine Vertical Line Count Register	**IMGDMA1_VDOENC_LB_WR_YCNT**
IMGDMA1+0280h	Video Decode DMA Start Register	**IMGDMA1_VDODEC_STR**
IMGDMA1+0284h	Video Decode DMA Control Register	**IMGDMA1_VDODEC_CON**
IMGDMA1+0290h	Video Decode DMA Y Base Address Register of Source Image	**IMGDMA1_VDODEC_Y_BASE**

IMGDMA1+0294h	Video Decode DMA U Base Address Register	**IMGDMA1_VDODEC_U_BASE**
IMGDMA1+0298h	Video Decode DMA V Base Address Register	**IMGDMA1_VDODEC_V_BASE**
IMGDMA1+02A0h	Video Decode DMA Horizontal Size Register	**IMGDMA1_VDODEC_XSIZE**
IMGDMA1+02A4h	Video Decode DMA Vertical Size Register of Source Image	**IMGDMA1_VDODEC_YSIZE**
IMGDMA1+02A8h	Video Decode DMA Pixel Number Register of Source Image	**IMGDMA1_VDODEC_PXLNUM**
IMGDMA1+02B0h	Video Decode DMA Y Component Horizontal Pixel Count Register	**IMGDMA1_VDODEC_Y_XCNT**
IMGDMA1+02B4h	Video Decode DMA Y Component Vertical Line Count Register	**IMGDMA1_VDODEC_Y_YCNT**
IMGDMA1+02B8h	Video Decode DMA V Component Horizontal Pixel Count Register	**IMGDMA1_VDODEC_V_XCNT**
IMGDMA1+02BCh	Video Decode DMA V Component Vertical Line Count Register	**IMGDMA1_VDODEC_V_YCNT**
IMGDMA1+0780h	Overlay DMA Multiple Output Engine 1 Start Register	**IMGDMA1_OVL_MO_1_STR**
IMGDMA1+0784h	Overlay DMA Multiple Output Engine 1 Control Register	**IMGDMA1_OVL_MO_1_CON**
IMGDMA1+0788h	Overlay DMA Multiple Output Engine 1 Busy Status	**IMGDMA1_OVL_MO_1_BUSY**
IMGDMA1+0C00h	Image Rotator 0 DMA Start Register	**IMGDMA1_IRT0_STR**
IMGDMA1+0C04h	Image Rotator 0 DMA Control Register	**IMGDMA1_IRT0_CON**
IMGDMA1+0C10h	Image Rotator 0 DMA Base Address Register of Line Buffer	**IMGDMA1_IRT0_FIFO_BASE**
IMGDMA1+0C14h	Image Rotator 0 DMA FIFO Length Register	**IMGDMA1_IRT0_FIFOYLEN**
IMGDMA1+0C20h	Image Rotator 0 DMA Horizontal Size Register of Source Image	**IMGDMA1_IRT0_XSIZE**
IMGDMA1+0C24h	Image Rotator 0 DMA Vertical Size Register of Source Image	**IMGDMA1_IRT0_YSIZE**
IMGDMA1+0C30h	Image Rotator 0 DMA Write Pointer	**IMGDMA1_IRT0_WRPTR**
IMGDMA1+0C40h	Image Rotator 0 DMA Read Pointer	**IMGDMA1_IRT0_RDPTR**

 © 2009 MediaTek Inc.

IMGDMA1+0C44h	Image Rotator 0 DMA Read Horizontal Pixel Count Register of Output Image	**IMGDMA1_IRT0_RDXCNT**
IMGDMA1+0C48h	Image Rotator 0 DMA Read Vertical Line Count Register of Output Image	**IMGDMA1_IRT0_RDYCNT**
IMGDMA1+0C50h	Image Rotator 0 DMA FIFO Line Count Register	**IMGDMA1_IRT0_FIFOCNT**
IMGDMA1+0C54h	Image Rotator 0 DMA FIFO Write Line Index Register	**IMGDMA1_IRT0_WRYIDX**
IMGDMA1+0C58h	Image Rotator 0 DMA FIFO Read Line Index Register	**IMGDMA1_IRT0_RDYIDX**
IMGDMA1+0C80h	Image Rotator 0 DMA Multiple Output Engine 1 Start Register	**IMGDMA1_IRT0_MO_1_STR**
IMGDMA1+0C84h	Image Rotator 0 DMA Multiple Output Engine 1 Control Register	**IMGDMA1_IRT0_MO_1_CON**
IMGDMA1+0C88h	Image Rotator 0 DMA Multiple Output Engine 1 Busy Status	**IMGDMA1_IRT0_MO_1_Busy**

Table 107 ***Image DMA 1 Register Map***

IMGDMA1+0000h Image DMA 1 Status Register IMGDMA1_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**IRT0 RUN**									**VDODEC RUN**	**VDOENCR RUN**	**VDOENCW RUN**	**JPEG RUN**
Type				RO									RO	RO	RO	RO
Reset				0									0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**IRT0 IT**									**VDODEC IT**	**VDOENCR IT**	**VDOENCW IT**	**JPEG IT**
Type				RO									RO	RO	RO	RO
Reset				0									0	0	0	0

This register helps software program being well aware of the global status of Image DMA 1 channels.

IT:

Interrupt status for engines inner Image DMA 1

0: No interrupt is generated by associated engine of Image DMA 1

1: An interrupt is generated by associated engine of Image DMA 1 and waiting for service. It would be cleared

after setting corresponding bit in IMGDMA1_ACKINT Register.

RUN:

DMA engine status

0: DMA engine is stopped or has completed the transfer already.

1: DMA engine is currently running. If the engine is set to be auto-restart, the related bits will be always 1 until the engine is turned off.

IMGDMA1+0004h Image DMA 1 Interrupt Acknowledge Register IMGDMA1_ACKINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				IRT0 ACK									VDODEC ACK	VDOENCR ACK	VDOENCW ACK	JPEG ACK
Type				WO									WO	WO	WO	WO

This register is used to acknowledge the current interrupt request associated with the completion event of a DMA channel by software program. Note that this is a write-only register, and any read to it will return a value of "0".

ACK:

Interrupt acknowledge for the DMA channel

0: No effect

1: Interrupt request of associated engine is acknowledged and interrupt status shown in IMGDMA1_STA should be relinquished.

IMGDMA1+0020h Image DMA 1 GMC Interface Status IMGDMA1_GMCIF_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								VDOENC RLB GMC BUSY	JPEG R GMC BUSY	IRT0 R GMC BUSY	VDOENC WLB GMC BUSY	JPEG W GMC BUSY	IRT0 W GMC BUSY	VDOENC RDMA GMC BUSY	VDODEC GMC BUSY	VDOENC WDMA GMC BUSY
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													IF3 BUSY	IF2 BUSY	IF1 BUSY	IF0 BUSY
Type													RO	RO	RO	RO

There are 9 GMC related engines and four GMC ports in Image DMA 1. This register is used to figure out which engine or which set of GMC interface is busy. When the engine or GMC interface is busy, it means that there are data waiting to read from / write into memory.

XXX GMC BUSY:

Engine busy signal.

0: Engine does not issue a command.

1: Engine issues a command and not completed yet.

IFX BUSY:

GMC Interface busy signal.

0: GMC interface is idle. No action occurs in this port.

1: GMC interface is busy. A command is not completed yet.

IMGDMA1+0100h JPEG DMA Start Register IMGDMA1_JPEG_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of JPEG DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for JPEG DMA.

0: stop JPEG DMA

1: activate JPEG DMA

IMGDMA1+0104h JPEG DMA Control Register IMGDMA1_JPEG_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**CAM_CRZ_VSYNC_EN**				**AUTO RSTR**	**MODE 1**	**MODE 0**	**IT**
Type									R/W				R/W	R/W	R/W	R/W
Reset									0				0	0	0	0

IT:

Interrupt Enabling

0: Disable

1: Enable

MODE:

JPEG DMA working mode. JPEG DMA receives YUV444 pixel and convert received pixel data into block-based

data and then output to JPEG Encoder according to mode setting here.

00: YUV422 mode. Convert to YUV422 block data.

01: Gray mode. Only Y component are converted to block data.

10: YUV420 mode. Convert to YUV420 block data.
11: YUV411 mode. Convert to YUV411 block data.

AUTO RSTR:

Automatic restart. JPEG DMA automatically restarts itself while current frame is finished.
0: Disable
1: Enable

CAM_CRZ_VSYNC_EN:

This control bit is an **ECO solution** for controlling the frame synchronous of OVL_MO_1 and VDOENC WDMA to camera and CRZ. Notice it should be configured **only when** OVL_MO_1 output to VDOENC WDMA (video preview / video capture scenario) .
0: Disable frame synchronous to camera and CRZ
1: Disable frame synchronous to camera and CRZ

IMGDMA1+0110h JPEG DMA Base Address Register of Line FIFO IMGDMA1_JPEG_FIFO_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADR[15:0]															
Type	R/W															

ADDR:

Base address of line FIFO used by JPEG DMA. ***It should align at 8x address***, or JPEG DMA will read wrong data.

Besides, it should be set in the internal memory to get better performance.

Note:

Notice that JPEG DMA is inside graph2sys, so the internal memory address would be better ranging from 0x4002_0000 to 0x4004_3fff than from 0x4000_0000 to 0x4001_7fff.

IMGDMA1+0114h JPEG DMA FIFO Length Register IMGDMA1_JPEG_FIFOLEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFOLEN															
Type	R/W															

FIFOLEN:

JPEG DMA FIFO Length. FIFOLEN must be the multiple of 8. Recommended values are at least 16 to support double buffer.

Note:

 © 2009 MediaTek Inc.

The FIFO size is determined by three parameter, JPEG_XSIZE, JPEG_FIFOLEN, and mode set in JPEG_CON.

In gray mode:

In gray mode, only Y component needs to be transmitted to JPEG Encoder. Because JPEG Encoder needs 8x8 block based data, JPEG DMA will pad horizontal size of source image to 8x. Therefore the realistic horizontal size used to calculate the FIFO size is

PAD_XSIZE_GRAY = ((JPEG_XSIZE + 7) >> 3) << 3

The byte per pixel in this mode is

BPP_GRAY = 1.

And thus

FIFO_SIZE = PAD_XSIZE_GRAY * JPEG_FIFOLEN * BPP_GRAY.

In YUV422 mode:

In YUV422 mode, U&V component both down-sampled 2x. Because JPEG Encoder needs 8x8 block based data of Y, U, and V component, respectively, JPEG DMA will pad horizontal size of source image to 16x to ensure U&V component transmitted to JPEG Encoder could be 8x. Therefore the realistic horizontal size used to calculate the FIFO size is

PAD_XSIZE_422 = ((JPEG_XSIZE + 15) >> 4) << 4

The byte per pixel is

BPP_422 = 2

And thus

FIFO_SIZE = PAD_XSIZE_422 * JPEG_FIFOLEN * BPP_422

In YUV411 mode:

In YUV411 mode, U&V component both down-sampled 4x. Because JPEG Encoder needs 8x8 block based data of Y, U, and V component, respectively, JPEG DMA will pad horizontal size of source image to 32x to ensure U&V component transmitted to JPEG Encoder could be 8x. Therefore the realistic horizontal size used to calculate the FIFO size is

PAD_XSIZE_411 = ((JPEG_XSIZE + 31) >> 5) << 5

The byte per pixel is

BPP_411 = 1.5

And thus

FIFO_SIZE = PAD_XSIZE_411 * JPEG_FIFOLEN * BPP_411

In YUV420 mode:

In YUV 420 mode, the vertical size of U/V component are also down-sampled 2x. In order to transmit

© 2009 MediaTek Inc.

8x8 block based data of U&V component to JPEG Encoder, JPEG DMA must collect 16 line data. So, if JPEG_FIFOLEN is set to 8, it would use 16 lines to store Y component data and thus 8 lines to store U&V component. Besides, as in YUV422 mode, the realistic horizontal size used to calculated the FIFO size is

PAD_XSIZE_420 = ((JPEG_XSIZE + 15) >> 4) << 4

The byte per pixel is

BPP_420 = 1.5

And thus

FIFO_SIZE = PAD_XSIZE_420 * JPEG_FIFOLEN * 2 * BPP_420

IMGDMA1+0120h JPEG DMA Horizontal Size Register of Source Image IMGDMA1_JPEG_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**XSIZE**															
Type	R/W															

XSIZE:

Horizontal size of the source frame. If the value is X, then it represents the source is an X-pixel wide image.

IMGDMA1+0124h JPEG DMA Vertical Size Register of Source Image IMGDMA1_JPEG_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**YSIZE**															
Type	R/W															

YSIZE:

Vertical size of the source frame. If the value is Y, then it represents the source is a Y-line high image.

IMGDMA1+0130h JPEG DMA Write Pointer Register IMGDMA1_JPEG_WRPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WRPTR[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WRPTR[15:0]**															
Type	RO															

WRPTR:

© 2009 MediaTek Inc.

Write pointer to display current writing address.

IMGDMA1+0134h JPEG DMA Write Horizontal Pixel Count Register of Received Image IMGDMA1_JPEG_WRXCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Horizontal pixel count of input image. It is a 16-bit up counter. If the value is X, then it represents JPEG DMA now

receives $(X+1)^{th}$ pixel of a line of the input image.

Besides, while it counts to (JPEG_XSIZE – 1), it will be reset to 0.

IMGDMA1+0138h JPEG DMA Write Vertical Line Count Register of Received Image IMGDMA1_JPEG_WRYCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Vertical line count of input image. It is a 16-bit up counter. If the value is Y, then it represents IBW1 DMA now at

$(Y+1)^{th}$ line of the input image.

Besides, it will increases 1 while JPEG_WRXCNT counts to (JPEG_XSIZE – 1) and JPEG DMA receives a pixel

successfully.

Note:

When JPEG DMA starts:

(JPEG_WRXCNT, JPEG_WRYCNT) = (0, 0);

When JPEG DMA finishes:

(JPEG_WRXCNT, JPEG_WRYCNT) = (0, JPEG_YSIZE);

IMGDMA1+0140h JPEG DMA Read Pointer Register IMGDMA1_JPEG_RDPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RDPTR[31:16]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RDPTR[15:0]															
Type	RO															

 © 2009 MediaTek Inc.

RDPTR:

Read pointer to display current reading address.

IMGDMA1+0144h JPEG DMA Read Horizontal Block Count Register of Padded Image IMGDMA1_JPEG_RDXBLK_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			COUNT													
Type			RO													

COUNT:

Display the YUV block count within a segment. The segment include 16 lines in YUV420 mode, and 8 lines in other mode.

It is a 14-bits down counter. If its value is C, it represents that there are still (C+1) 8x8 block of current segment not read out of FIFO completely, including current reading block.

If all 8x8 blocks of current segment are read out of FIFO, then it will be reset to its initial value.

Note:

Initial value of JPEG_RDXBLK_CNT is as followed.

In gray mode:

The total Y component block within a segment (8 lines) is

Y_BLOCK_NUM_GRAY = PAD_XSIZE_GRAY/8

So the initial value of this register under this mode is

JPEG_RDXBLK_CNT = Y_BLOCK_NUM_GRAY - 1

In YUV422 mode:

The total Y component block within a segment (8 lines) is

Y_BLOCK_NUM_422 = PAD_XSIZE_422/8

And due to U&V component being horizontally down-sampled 2x,

U_BLOCK_NUM_422 = V_BLOCK_NUM_422 = PAD_XSIZE_422/16

So the initial value of this register under this mode is

JPEG_RDXBLK_CNT = (Y_BLOCK_NUM_422+U_BLOCK_NUM_422+V_BLOCK_NUM_422) - 1

In YUV411 mode:

The total Y component block within a segment (8 lines) is

Y_BLOCK_NUM_411 = PAD_XSIZE_411/8

And due to U&V component being horizontally down-sampled 4x,

U_BLOCK_NUM_411 = V_BLOCK_NUM_411 = PAD_XSIZE_411/32

So the initial value of this register under this mode is

JPEG_RDXBLK_CNT = (Y_BLOCK_NUM_411+U_BLOCK_NUM_411+V_BLOCK_NUM_411) - 1

© 2009 MediaTek Inc.

In YUV420 mode:

The total Y component block within a segment (16 lines) is

Y_BLOCK_NUM_420 = (PAD_XSIZE_420 / 8) * 2

And due to U&V component being horizontally down-sampled 2x,

U_BLOCK_NUM_420 = V_BLOCK_NUM_420 = PAD_XSIZE_420 / 16

So the initial value of this register under this mode is

JPEG_RDXBLK_CNT = (Y_BLOCK_NUM_420+U_BLOCK_NUM_420+V_BLOCK_NUM_420) - 1

IMGDMA1+0148h JPEG DMA Read Vertical Block Count Register of Padded Image IMGDMA1_JPEG_RDYBLK_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				COUNT												
Type				RO												

COUNT:

Display the vertical segment count of padding image. The segment includes 16 lines in YUV420 mode and 8 lines in other mode.

It is a 13-bits down counter. If its value is C, it represents that JPEG DMA is current at $(C+1)^{th}$ segments counting from bottom of the padding image.

It will be decreased by 1 each time JPEG DMA reads all 8x8 blocks of current segment out of FIFO, except JPEG DMA is currently at the bottom segment of padding image, i.e., JPEG_RDBLK_CNT is 0 now.

Note:

The initial value of JPEG_RDYBLK_CNT is as followed.

In YUV420 mode:

The vertical segment of padding image in the mode is

PAD_Y_SEG_420 = (JPEG_YSIZE + 15)>>4

So the initial value is

JPEG_RDYBLK_CNT = PAD_Y_SEG_420 – 1

In other mode:

The vertical segment of padding image in the mode is

PAD_Y_SEG_NON_420 = (JPEG_YSIZE + 7)>>3

So the initial value is

JPEG_RDYBLK_CNT = PAD_Y_SEG_NON_420 – 1

Note:

When JPEG DMA finishes, (JPEG_RDXBLK_CNT, JPEG_RDYBLK_CNT) are always equal to (0, 0)

IMGDMA1+0150h JPEG DMA FIFO Line Count Register IMGDMA1_JPEG_FFCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COUNT															
Type	RO															

COUNT:

Display how many lines are stored in FIFO now. It could count up or count down.

Count up:

In YUV420 mode, because U&V component are down-sampled 2x, the counter will increase 1 while 2 lines of source image are written into FIFO completely; in other mode, the counter will increase 1 while 1 line of source image is writing into FIFO completely.

Therefore, in YUV420 mode, if its value is C, it represents there are (C*2) lines of source image are stored in FIFO now; in other mode, a value C represents there are C lines of source image is stored in FIFO now.

Count down:

While JPEG DMA read all 8x8 blocks of a segment (16 lines in YUV420 mode, and 8 lines in other mode), it will decrease by 8.

IMGDMA1+0154h JPEG DMA FIFO Write Line Index Register IMGDMA1_JPEG_FFWRLIDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YIDX															
Type	RO															

YIDX:

Display which FIFO line JPEG DMA is now writing. If its value is Y, then JPEG DMA is now writing Y^{th} FIFO line.

In YUV420 mode, because U&V component are down-sampled 2x, the counter will increase 1 while 2 lines of source are written into FIFO completely; in other mode, the counter will increase 1 while 1 line of source image is writing into FIFO completely.

When its value is equal to JPEG_FIFOLEN (at the bottom of the FIFO) and writing a FIFO line is completed, it will

be reset to 1 to acknowledge JPEG DMA to write to top of the FIFO again.

.

IMGDMA1+0158h JPEG DMA FIFO Read Line Index Register IMGDMA1_JPEG_FFRDLIDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**YIDX**												
Type				RO												

YIDX:

Display which FIFO segment (8 FIFO line) JPEG DMA is now reading. If its value is Y, then JPEG DMA is now

reading Y[th] FIFO segment.

When its value is equal to JPEG_FIFOLEN>>3 (at the bottom of the FIFO segment) and all 8x8 blocks of current

FIFO segment are read out, it will be reset to 1 to acknowledge JPEG DMA to read from top of the FIFO again.

IMGDMA1+0200h Video Encode DMA Start Register IMGDMA1_VDOENC_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of VDOENC DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for VDOENC DMA.

0: stop VDOENC DMA

1: activate VDOENC DMA

IMGDMA1+0204h Video Encode DMA Control Register IMGDMA1_VDOENC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									VDOENC_ULTRA_HIGH_EN	FLIP	ROT	RD IT	AUTO RSTR	W2R	WR IT
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									0	0	0	0	0	0	0

WR IT:

WDMA Done Interrupt Enable. Interrupt issues when all of the transfers are done. For auto-restart mode, interrupt issues at every restart.

0: Disable

1: Enable

W2R:

WDMA triggers RDMA. While this function is enabled, VDOENC WDMA will write data to video buffer first, and then a start pulse will issue to VDOENC RDMA. If VDOENC RDMA is idle now, it will accept the start pulse to read back the same buffer; else, it will ignore the start pulse until its current job is completed.

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. VDO DMA automatically restarts while current frame is finished. Base address will be automatically switched between VDOENC_Y/U/V_BASE1 and VDOECN_Y/U/V_BASE2.

0: Disable

1: Enable

RD IT:

RDMA Done Interrupt Enable. Interrupt issues when all of the transfers are done.

0: Disable

1: Enable

ROT:

Rotation direction related to source frame buffer. VDOENC RDMA must co-work with IRT0 DMA to execute the function.

00: No rotation

01: 90° rotation

10: 180° rotation

11: 270° rotation

FLIP:

Flipping option related to source frame buffer. VDOENC RDMA must co-work with IRT0 DMA to execute the function.

0: No flip

1: Flipped after rotation

VDOENC_ULTRA_HIGH_EN:

This control bit is an **ECO solution** for enabling VDOENC WDMA ultra high function. When this port is enabled, VDOENC WDMA will get more bandwidth from external memory.

0: Disable VDOENC ultra high function

1: Enable VDOENC ultra high function

Note:

VDOENC RDMA ***must*** co-work with IRT0 to achieve rotation and/or flip function. Therefore the rotation/flip setting of VDOENC DMA & IRT0 DMA must be the same.

IMGDMA1+0210h Video Encode DMA Y Base Address 1 Register IMGDMA1_VDOENC_Y_BASE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the Y component of destination frame buffer 1. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

IMGDMA1+0214h Video Encode DMA U Base Address 1 Register IMGDMA1_VDOENC_U_BASE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the U component of destination frame buffer 1. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

IMGDMA1+0218h Video Encode DMA V Base Address 1 Register IMGDMA1_VDOENC_V_BASE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

© 2009 MediaTek Inc.

ADDR:

Base address of the V component of destination frame buffer 1. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

IMGDMA1+0220h Video Encode DMA Y Base Address 2 Register IMGDMA1_VDOENC_Y_BASE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the Y component of destination frame buffer 2. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

By the way, this register is only effective when VDOENC DMA is working ***under auto-restart mode***.

IMGDMA1+0224h Video Encode DMA U Base Address 2 Register IMGDMA1_VDOENC_U_BASE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the U component of destination frame buffer 2. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

By the way, this register is only effective when VDOENC DMA is working ***under auto-restart mode***.

IMGDMA1+0228h Video Encode DMA V Base Address 2 Register IMGDMA1_VDOENC_V_BASE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the V component of destination frame buffer 2. ***It should align at 8x address***, or VDOENC DMA will read/write wrong data.

By the way, this register is only effective when VDOENC DMA is working ***under auto-restart mode***.

IMGDMA1+0230h Video Encode DMA Horizontal Size Register of Source Image IMGDMA1_VDOENC_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					XSIZE											
Type					R/W											

XSIZE:

Horizontal size of the source image. If the value is X, then it represents the source is an X-pixel wide image.

Note that the horizontal size of the source image must be 16x.

IMGDMA1+0234h Video Encode DMA Vertical Size Register of Source Image IMGDMA1_VDOENC_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					YSIZE											
Type					R/W											

YSIZE:

Vertical size of the source image. If the value is Y, then it represents the source is a Y-line high image.

Note that the vertical size of the source image must be 16x.

IMGDMA1+0238h Video Encode DMA Pixel Number Register of Source Image IMGDMA1_VDOENC_PXLNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									NUM							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	NUM															
Type	R/W															

NUM:

Pixel number of the source image. If its value is N, then it represents the source is an N-pixel image.

Note pixel number of the frame is equal to VDOENC_XSIZE * VDOENC_YSIZE.

IMGDMA1+0240h Video Encode Write DMA Horizontal Pixel Count Register of Written Image IMGDMA1_VDOENC_WXCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name					COUNT	CUR_FRAME
Type					RO	RO

COUNT:

After ECO, we use (COUNT*2) to show the currently horizontal pixel position of a segment (4 lines) of Y frame

buffer where VDOENC WDMA is now at. If its value is C, then it represents VDOENC WDMA now at $(C-7)^{th}$ ~

C^{th} pixel of a segment of Y frame buffer.

It is a up-counter. If VDOENC WDMA writes last 8 pixel data of a segment, it will be reset to 8 to show VDOENC

WDMA will be at first 8 pixel of next segment.

CUR_FRAME:

To represent the current working frame of ***VDOENC WDMA.***

IMGDMA1+0244h Video Encode Write DMA Vertical Line Count Register of Written Image IMGDMA1_VDOENC_WYCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Show the currently vertical line position of Y frame buffer where VDOENC WDMA is at. If its value is C,

then it represents VDOENC WDMA is now at $(C-3)^{th}$ ~ C^{th} line (a segment) of Y frame buffer.

It is a up-counter. If VDOENC WDMA writes out last 8 pixel data of a segment, it will be increased by 4 to show

VDOENC WDMA switches to next segment.

Note:

When VDOENC DMA starts:

(VDOENC_WXCNT, VDOENC_WYCNT) = (8, 4)

When VDOENC DMA finishes:

(VDOENC_WXCNT, VDOENC_WYCNT) = (8, VDOENC_YSIZE + 4)

IMGDMA1+0250h Video Encode Read DMA Y component Horizontal Pixel Count Register IMGDMA1_VDOENC_Y_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Show the currently horizontal pixel position of a segment (4 lines) of Y frame buffer where VDOENC RDMA is now

at. If its value is C, then it represents VDOENC RDMA is now at (C-3)th ~ Cth pixel data of a segment of Y frame

buffer.

It is a up-counter, and it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 when VDOENC_Y_YCNT counts to VDOENC_YSIZE and 16 bytes data of current

position are read out of Y frame buffer.

When rotation is 0° or 180° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will

be reset to 4 while it counts to VDOENC_XSIZE and 16 bytes data are read out.

Note:

The pixel index is counted from either left or right according to the setting of rotation and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Left	Right
90	Left	Left
180	Right	Left
270	Right	Right

IMGDMA1+0254h Video Encode Read DMA Y Component Vertical Line Count Register IMGDMA1_VDOENC_Y_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Show the currently vertical line position of Y frame buffer where VDOENC RDMA is now at. If its value is C, then it

represents VDOENC RDMA is now at (C-3)th ~ Cth line (a segment) of Y frame buffer.

It is a up-counter, and it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will

be reset to 4 while it counts to VDOENC_YSIZE and 16 bytes data are read out.

When rotation is 0° or 180° :

It is increased by 4 when VDOENC_Y_XCNT counts to VDOENC_XSIZE and 16 bytes data of current position are read out of Y frame buffer.

Note:

The line index is counted from either top or bottom according to the setting of rotation and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Top	Top
90	Bottom	Top
180	Bottom	Bottom
270	Top	Bottom

Note:

VDOENC RDMA accept a start pulse from VDOENC WDMA

(VDOENC_Y_XCNT, VDOENC_Y_YCNT) = (4 , 4);

VDOENC RDMA finishes

When rotation is 90° or 270° :

(VDOENC_Y_XCNT, VDOENC_Y_YCNT) = (VDOENC_XSIZE + 4 , 4);

When rotation is 0° or 180° :

(VDOENC_Y_XCNT, VDOENC_Y_YCNT) = (4 , VDOENC_YSIZE + 4);

IMGDMA1+0258h Video Encode Read DMA V component Horizontal Pixel Count Register IMGDMA1_VDOENC_V_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						COUNT										
Type						RO										

COUNT:

Show the currently horizontal pixel position of a segment (4 lines) of V frame buffer where VDOENC RDMA is now at. If its value is C, then it represents VDOENC RDMA is now $(C-3)^{th}$ ~ C^{th} pixel data of a segment of V frame buffer.

It is a up-counter, and it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 when VDOENC_V_YCNT counts to VDOENC_YSIZE[11:1] and 16 bytes data of current position are read out of V frame buffer.

When rotation is 0° or 180° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will be reset to 4 while it counts to VDOENC_XSIZE and 16 bytes data are read out.

Note:

The pixel index is counted from either left or right according to the setting of rotation and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Left	Right
90	Left	Left
180	Right	Left
270	Right	Right

IMGDMA1+025Ch Video Encode Read DMA V Component Vertical Line Count Register IMGDMA1_VDOENC_V_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name						COUNT										
Type						RO										

COUNT:

Show the currently vertical line position of V frame buffer where VDOENC RDMA is at. If its value is C, then it

represents VDOENC RDMA is now at $(C-3)^{th}$ ~ C^{th} line (a segment) of V frame buffer.

It is a up-counter, and it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 each time 16 bytes data of current position are read out of V frame buffer. And it will

be reset to 4 while it counts to VDOENC_YSIZE[11:1] and 16 bytes data are read out.

When rotation is 0° or 180° :

It is increased by 4 when VDOENC_V_XCNT counts to VDOENC_XSIZE[11:1] and 16 bytes data of

current position are read out of V frame buffer.

Note:

The line index is counted from either top or bottom according to the setting of rotation and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Top	Top
90	Bottom	Top
180	Bottom	Bottom
270	Top	Bottom

Note:

VDOENC RDMA accept a start pulse from VDOENC WDMA

(VDOENC_V_XCNT, VDOENC_V_YCNT) = (4 , 4);

VDOENC RDMA finishes

When rotation is 90° or 270° :

(VDOENC_V_XCNT, VDOENC_V_YCNT) = (VDOENC_XSIZE[11:1] + 4 , 4);

When rotation is 0° or 180° :

(VDOENC_V_XCNT, VDOENC_V_YCNT) = (4 , VDOENC_YSIZE[11:1] + 4);

IMGDMA1+0260h Video Encode Read DMA Dropped Frame Count Register IMGDMA1_VDOENC_DROP_FCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COUNT**															
Type	RO															

COUNT:

VDOENC RDMA dropped frame drop counter. When VDOENC WDMA finished writing a frame to memory and issued a start pulse to VDOENC RDMA, VDOECN RDMA would accept or ignore the pulse depending on it is busy or idle now. If it is busy, then VDOENC RDMA will ignore the pulse and the dropped frame counter will be increased by 1.

Note:

The counter will be reset when VDOENC_STR is set to 0.

IMGDMA+0270h Video Encode DMA Line Buffer Base Address Register IMGDMA1_VDOENC_LB_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of line buffer of video frame. The purpose of using line buffers is to convert scan-line data into 4x4 block data before writing data into frame buffer. ***It should align at 8x address***, or VDOENC DMA will read wrong data. Besides, it should be set in the internal memory to get better performance.

Note:

Notice that VDOENC DMA is inside graph2sys, so the internal memory address would be better ranging from 0x4002_0000 to 0x4004_3fff than from 0x4000_0000 to 0x4001_7fff.

IMGDMA+0274h Video Encode DMA Line Buffer Length Register IMGDMA1_VDOENC_LB_YLEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					LB_YLEN											
Type					R/W											

LB_YLEN:

Line buffer length of VDOENC DMA Line Buffer Write Engine. ***It must be 4x***. The line buffer size is calculated as

followed.

Line Buffer Size = VDOENC_XSIZE * LB_YLEN * 2 bytes

IMGDMA+0278h Video Encode DMA Line Buffer Write Engine Horizontal Pixel Count Register IMGDMA11_VDOENC_LB_WR_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Horizontal pixel count of input image. It is a 12-bit up counter. If the value is C, then it represents VDOENC DMA

now writes $(C-3)^{th}$ ~ C^{th} pixel into line buffer. While it counts to VDOENC_XSIZE[11:2] and the write-line-buffer

command is finished, it will be reset to 4.

IMGDMA+027Ch Video Encode DMA Line Buffer Write Engine Vertical Line Count Register IMGDMA1_VDOENC_LB_WR_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Vertical line count of input image. It is a 12-bit up counter. If the value is C, then it represents VDOENC DMA is

now at $(C-3)^{th}$ ~ C^{th} line of input image. While VDOECN_LB_WR_XCNT will be reset to 4, this counter will

increase 1 at the same time.

Note:

VDOENC DMA starts:

(VDOENC_LB_WR_XCNT, VDOENC_LB_WR_XCNT) = (4, 1)

VDOENC DMA finishes:

(VDOENC_LB_WR_XCNT, VDOENC_LB_WR_XCNT) = (4, VDOENC_YSIZE + 1)

IMGDMA1+0280h Video Decode DMA Start Register IMGDMA1_VDODEC_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of VDODEC DMA. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for VDODEC DMA.

0: stop VDODEC DMA

1: activate VDODEC DMA

IMGDMA1+0284h Video Decode DMA Control Register IMGDMA1_VDODEC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SCAN MODE**	**FLIP**	**ROTATION**					**IT**
Type									R/W	R/W	R/W					R/W
Reset									0	0	0					0

IT:

Interrupt Enabling. Interrupt of VDODEC DMA issues when all the data transmission of an MP4 frame are done.

0: Disable

1: Enable

ROT:

Rotation direction related to source frame buffer. VDODEC DMA must co-work with IRT0 DMA to execute the

function.

00: No rotation

01: 90° rotation

10: 180° rotation

11: 270° rotation

FLIP:

Flipping option related to source frame buffer. VDODEC DMA must co-work with IRT0 DMA to execute the
function.

0: No flip

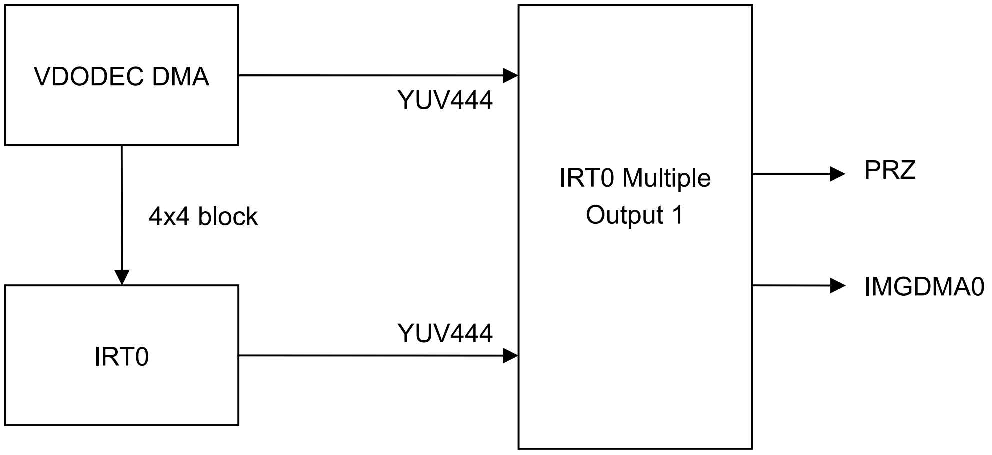
1: Flipped after rotation

SCAN MODE:

To indicate VDODEC DMA the frame buffer is scan-line based or 4x4 block based. If VDODEC DMA is in scan line
mode, no rotation or flip function are valid. And it will transmit ***pixel data*** to IRT0 Multiple Output 1 instead transmit
4x4 block data to IRT0 DMA.

0: 4x4 block based

1: Scan-line based



IMGDMA1+0290h **Video Decode DMA Y Base Address Register** **IMGDMA1_VDODEC_Y_BASE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the Y component of source frame buffer. ***It should align at 8x address***, or
VDODEC DMA will read

 © 2009 MediaTek Inc.

wrong data.

IMGDMA1+0294h Video Decode DMA U Base Address Register IMGDMA1_VDODEC_U_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the U component of source frame buffer. ***It should align at 8x address***, or VDODEC DMA will read
wrong data.

IMGDMA1+0298h Video Decode DMA V Base Address Register IMGDMA1_VDODEC_V_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR:

Base address of the V component of source frame buffer. ***It should align at 8x address***, or VDODEC DMA will read
wrong data.

IMGDMA1+02A0h Video Decode DMA Horizontal Size Register of Source Image IMGDMA1_VDODEC_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							XSIZE									
Type							R/W									

XSIZE:

Horizontal size of the source image. If the value is X, then it represents the source is an X-pixel wide image.

Note that the horizontal size of the source image must be 16x.

IMGDMA1+02A4h Video Decode DMA Vertical Size Register of Source Image IMGDMA1_VDODEC_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**YSIZE**									
Type							R/W									

YSIZE:

Vertical size of the source image. If the value is Y, then it represents the source is a Y-line high image.
Note that the vertical size of the source image must be 16x.

IMGDMA1+02A8h Video Decode DMA Pixel Number Register of Source Image IMGDMA1_VDODEC_PXLNUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**NUM**			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**NUM**															
Type	R/W															

NUM:

Pixel number of the source image. If its value is N, then it represents the source is an N-pixel image.
*Note pixel number of the frame is equal to VDODEC_XSIZE * VDODEC_YSIZE*.

IMGDMA1+02B0h Video Decode DMA Y component Horizontal Pixel Count Register IMGDMA1_VDODEC_Y_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COUNT**									
Type							RO									

COUNT:

Show the currently horizontal pixel position of a segment (4 lines) of Y frame buffer where VDODEC DMA is now
at. If its value is C, then it represents VDODEC DMA is now at (C-3)th ~ Cth pixel data of a segment of Y frame
buffer.

It is a up-counter, and it works at three modes.

Under scan-line mode:

It is increased by 16 each time 16 bytes data of current position are read out of Y frame buffer. And it will
be reset to 16 while it counts to VDODEC_XSIZE and 16 bytes data are read out.

Under 4x4 block mode & rotation is 90° or 270° :

It is increased by 4 when VDODEC_Y_YCNT counts to VDODEC_YSIZE and 16 bytes data of current
position are read out of Y frame buffer.

Under 4x4 block mode & rotation is 0° or 180° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will be reset to 4 while it counts to VDODEC_XSIZE and 16 bytes data are read out.

Note:

When under scan-line mode, the pixel index is always counted from left.

When under 4x4 block mode, the pixel index is counted from either left or right according to the setting of rotation and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Left	Right
90	Left	Left
180	Right	Left
270	Right	Right

IMGDMA1+02B4h Video Decode DMA Y Component Vertical Line Count Register IMGDMA1_VDODEC_Y_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							COUNT									
Type							RO									

COUNT:

Show the currently vertical line position of Y frame buffer where VDODEC DMA is now at. If its value is C, then it represents VDODEC DMA is now at $(C-3)^{th}$ ~ C^{th} line (a segment) of Y frame buffer.

It is a up-counter, and it works at three modes.

Under scan-line mode

It is increased by 1 each time VDODEC_Y_XCNT count to VDODEC_XSIZE and 16 bytes data of current position are read out of Y frame buffer.

Under 4x4 block mode & rotation is 90° or 270° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will be reset to 4 while it counts to VDODEC_YSIZE and 16 bytes data are read out.

Under 4x4 block mode & rotation is 0° or 180° :

It is increased by 4 when VDODEC_Y_XCNT counts to VDODEC_XSIZE and 16 bytes data of current position are read out of Y frame buffer.

Note:

When under scan-line mode, the line index is always counted from top.

When under 4x4 block mode, the line index is counted from either top or bottom according to the setting of rotation
and flip bits in VDOENC_CON.

Rotation \ Flip	0	1
0	Top	Top
90	Bottom	Top
180	Bottom	Bottom
270	Top	Bottom

Note:

VDODEC DMA starts

Under scan-line mode

(VDODEC_Y_XCNT, VDODEC_Y_YCNT) = (16 , 1);

Under 4x4 block mode

(VDOENC_Y_XCNT, VDOENC_Y_YCNT) = (4 , 4);

VDODEC DMA finishes

Under scan-line mode:

(VDODEC_Y_XCNT, VDODEC_Y_YCNT) = (16 , VDODEC_YSIZE + 1);

Under 4x4 block mode & rotation is 90° or 270° :

(VDODEC_Y_XCNT, VDODEC_Y_YCNT) = (VDODEC_XSIZE + 4 , 4);

Under 4x4 block mode & rotation is 0° or 180° :

(VDODEC_Y_XCNT, VDODEC_Y_YCNT) = (4 , VDODEC_YSIZE + 4);

IMGDMA1+02B8h Video Decode DMA V component Horizontal Pixel Count Register IMGDMA1_VDODEC_V_XCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								COUNT								
Type								RO								

COUNT:

Show the currently horizontal pixel position of a segment (4 lines) of V frame buffer where VDODEC DMA is now
at. If its value is C, then it represents VDODEC DMA is now (C-3)th ~ Cth pixel data of a segment of V frame buffer.

It is a up-counter, but it is always 0 under scan-line mode. Under 4x4 block mode, it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 when VDOENC_V_YCNT counts to VDOENC_YSIZE[11:1] and 16 bytes data of
current position are read out of V frame buffer.

When rotation is 0° or 180° :

It is increased by 4 each time 16 bytes data of current position are read out of Y frame buffer. And it will be reset to 4 while it counts to VDOENC_XSIZE and 16 bytes data are read out.

Note:

The pixel index is counted from either left or right according to the setting of rotation and flip bits in VDODEC_CON.

Rotation \ Flip	0	1
0	Left	Right
90	Left	Left
180	Right	Left
270	Right	Right

IMGDMA1+02BCh Video Decode DMA V Component Vertical Line Count Register IMGDMA1_VDODEC_V_YCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								COUNT								
Type								RO								

COUNT:

Show the currently vertical line position of V frame buffer where VDODEC DMA is at. If its value is C, then it represents VDODEC DMA is now at $(C-3)^{th}$ ~ C^{th} line (a segment) of V frame buffer.

It is a up-counter, but it is always 0 under scan-line mode. Under 4x4 block mode, it works at two modes.

When rotation is 90° or 270° :

It is increased by 4 each time 16 bytes data of current position are read out of V frame buffer. And it will be reset to 4 while it counts to VDODEC_YSIZE[9:1] and 16 bytes data are read out.

When rotation is 0° or 180° :

It is increased by 4 when VDODEC_V_XCNT counts to VDODEC_XSIZE[9:1] and 16 bytes data of current position are read out of V frame buffer.

Note:

The line index is counted from either top or bottom according to the setting of rotation and flip bits in VDODEC_CON.

Rotation \ Flip	0	1
0	Top	Top
90	Bottom	Top
180	Bottom	Bottom
270	Top	Bottom

Note:

VDODEC DMA starts (4x4 block mode):

(VDODEC_V_XCNT, VDODEC_V_YCNT) = (4 , 4);

VDODEC DMA finishes (4x4 block mode)

When rotation is 90° or 270° :

(VDODEC_V_XCNT, VDODEC_V_YCNT) = (VDOENC_XSIZE[9:1] + 4 , 4);

When rotation is 0° or 180° :

(VDODEC_V_XCNT, VDODEC_V_YCNT) = (4 , VDOENC_YSIZE[9:1] + 4);

IMGDMA1+0780h Overlay DMA Multiple Output Engine 1 Start Register IMGDMA1_OVL_MO_1_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of OVL Multiple Output 1. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for OVL Multiple Output 1

0: stop OVL Multiple Output 1

1: activate OVL Multiple Output 1

Note:

OVL Multiple Output module is used to transmit OVL DMA output pixel data to multiple destinations simultaneously, including JPEG DMA, VDOENC WDMA, PRZ, DRZ, and Y2R0. Due to its destination located at Graphsys1 and Graphsys2 respectively, in order to transmit pixel data to different destinations located at different Graphsys, OVL Multiple Output has been divided into OVL Multiple Output 0 and OVL Multiple Output 1 to support multiple output function. ***The starting sequence of OVL DMA, OVL Multiple Output 0 and OVL Multiple Output 1 should be as followed***.

OVL Multiple Output 1 started → OVL Multiple Output 0 started → OVL DMA

 © 2009 MediaTek Inc.

If OVL DMA is allowed to transmit pixel data before OVL Multiple Output 0/1 are started, it would lose some pixels.

IMGDMA1+0784h Overlay DMA Multiple Output Engine 1 Control Register IMGDMA1_OVL_MO_1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRZ	VDOENC DMA	JPEG DMA
Type														R/W	R/W	R/W
Reset														0	0	0

JPEG DMA:

Enable output to JPEG DMA.

0: Disable

1: Enable

VDOENC DMA:

Enable output to VDOENC DMA.

0: Disable

1: Enable

PRZ:

Enable output to PRZ.

0: Disable

1: Enable

IMGDMA1+0788h Overlay DMA Multiple Output Engine 1 Busy Status IMGDMA1_OVL_MO_1_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRZ BUSY	VDOENC DMA BUSY	JPEG DMA BUSY
Type														RO	RO	RO

The BUSY bits are used to show if OVL Multiple Output 1 transmits data to destination or not. If there are two or more output path are enabled and only some BUSY bit stays at 1, then the OVL Multiple Output 1 may be stalled by the path.

JPEG DMA BUSY:

0: No data transmit to JPEG DMA now.

1: Data are transmitted to JPEG DMA 1 now.

VDOENC DMA BUSY:

0: No data transmit to VDOENC DMA now.

1: Data are transmitted to VDOENC DMA now.

PRZ BUSY:

0: No data transmit to PRZ now.

1: Data are transmitted to PRZ now.

IMGDMA1+0C00h Image Rotator 0 DMA Start Register IMGDMA1_IRT0_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IRT0 DMA. Note that before setting STR to "1", all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IRT0 DMA.

0: stop IRT0 DMA

1: activate IRT0 DMA

IMGDMA1+0C04h Image Rotator 0 DMA Control Register IMGDMA1_IRT0_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**FLIP**	**ROT**			**SRCSEL**	**AUTO RSTR**	**IT**
Type										R/W	R/W			R/W	R/W	R/W
Reset										0	0			0	0	0

IT:

IRT0 done interrupt enable. Interrupt issues when all data are transmitted to IRT0 Multiple Output 1. For auto-restart mode, interrupt issues at every restart.

0: Disable

1: Enable

AUTO RSTR:

Automatic restart. IRT0 DMA automatically restarts while current frame is finished.

0: Disable

1: Enable

SRCSEL:

Select source of IRT0 DMA.

0: VDOENC RDMA

1: VDODEC DMA. VDODEC DMA could transmit data to IRT0 DMA if and only if it works under 4x4 block mode.

ROT:

Rotation direction related to source frame buffer of VDOENC RDMA or VDODEC DMA.

00: No rotation

01: 90° rotation

10: 180° rotation

11: 270° rotation

FLIP:

Flip option related to source frame buffer of VDOENC RDMA or VDODEC DMA.

0: No flip

1: Flipped after rotation

Note:

IRT0 DMA ***must*** co-work with VDOENC RDMA or VDODEC DMA to achieve rotation and/or flip function.

Therefore the rotation/flip setting of IRT0 DMA & VDOENC RDMA/VDODEC DMA must be the same.

IMGDMA1+0C10h Image Rotator 0 DMA Base Address Register of Line Buffer IMGDMA1_IRT0_FIFO_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR:

Base address of IRT0 line buffer. The purpose of using line buffers is to convert 4x4 block based data into scan-line

data before writing data to IRT0 Multiple Output 1. ***It should align at 8x address***, or IRT0 DMA will read/write

wrong data. Besides, it should be set in the internal memory to get better performance.

Notice that IRT0 DMA is inside graph2sys, so the internal memory address would be better ranging from

0x4002_0000 to 0x4004_3fff than from 0x4000_0000 to 0x4001_7fff.

IMGDMA1+0C14h Image Rotator 0 DMA FIFO Length Register IMGDMA1_IRT0_FIFOYLEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name		FIFOLEN
Type		R/W

FIFOLEN:

IRT0 DMA FIFO Length. FIFOLEN must be the multiple of 8. Recommended values are at least 16 to support double buffer.

Note:

The FIFO size is determined by four parameter, IRT0_XSIZE, IRT0_YSIZE, IRT0_FIFOLEN, and rotation set in IRT0_CON.

When rotation is 90° or 270° :

FIFO Size = ***IRT0_YSIZE*** * IRT0_FIFOLEN * 1.5;

When rotation is 0° or 180° :

FIFO Size = ***IRT0_XSIZE*** * IRT0_FIFOLEN * 1.5;

IMGDMA1+0C20h Image Rotator 0 DMA Horizontal Size Register of Source Image IMGDMA1_IRT0_XSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					XSIZE											
Type					R/W											

XSIZE:

Horizontal size of the source image. If the value is X, then it represents the source is an X-pixel wide image.

Note that the horizontal size of the source image must be 16x.

IMGDMA1+0C24h Image Rotator 0 DMA Vertical Size Register of Source Image IMGDMA1_IRT0_YSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					YSIZE											
Type					R/W											

YSIZE:

Vertical size of the source image. If the value is Y, then it represents the source is a Y-line high image.

Note that the vertical size of the source image must be 16x.

IMGDMA1+0C30h Image Rotator 0 DMA Write Pointer IMGDMA1_IRT0_WRPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WRPTR[31:16]															

Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WRPTR[15:0]**															
Type	RO															

WRPTR:

Write pointer to display current writing address.

IMGDMA1+0C40h Image Rotator 0 DMA Read Pointer IMGDMA1_IRT0_RDPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RDPTR[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RDPTR[15:0]**															
Type	RO															

RDPTR:

Read pointer to display current reading address.

IMGDMA1+0C44h Image Rotator 0 DMA Read Horizontal Pixel Count Register of Output Image IMGDMA1_IRT0_RDXCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**COUNT**											
Type					RO											

COUNT:

Display the currently horizontal position of a line of Y component.

It is a 12-bits up counter. If its value is C, it represents IRT0 DMA now reads (C-7)th ~ Cth pixel data of a line of Y

component. If all pixel data of a line are read out, then the counter will be reset to its initial value.

Note:

When rotation is 90° or 270° :

It is increased by 8 each time 8 bytes data of current position are read out of Y frame buffer. And it will be reset

to 8 while it counts to IRT0_YSIZE and 8 bytes data are read out.

When rotation is 0° or 180° :

It is increased by 8 each time 8 bytes data of current position are read out of Y frame buffer. And it will be reset

to 8 while it counts to IRT0_XSIZE and 8 bytes data are read out.

IMGDMA1+0C48h Image Rotator 0 DMA Read Vertical Line Count Register of Output Image IMGDMA1_IRT0_RDYCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Display the currently vertical position of Y component.

It is a 12-bits up counter. If its value is C, it represents IRT0 DMA is now at C[th] line of Y component. If all pixel data

of a line are read out, the counter will be increased by 1.

Note:

IRT0 DMA starts

(IRT0_RDXCNT, IRT0_RDYCNT) = (8, 1)

IRT0 DMA finishes

When rotation is 90° or 270°,

(IRT0_RDXCNT, IRT0_RDYCNT) = (8, IRT0_XSIZE + 1)

When rotation is 0° or 180°,

(IRT0_RDXCNT, IRT0_RDYCNT) = (8, IRT0_YSIZE + 1)

IMGDMA1+0C50h Image Rotator 0 DMA FIFO Line Count Register IMGDMA1_IRT0_FIFOCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COUNT											
Type					RO											

COUNT:

Display how many lines are stored in FIFO now. It could count up or count down.

Count up:

Because the data written into FIFO are 4x4 block, IRT0 DMA will increase the counter 4 after it finishes

writing 4 line Y component. Therefore if its value is C, it represents there are C lines data in the FIFO now.

Count down:

IRT0 DMA will decrease the counter 4 while it reads out 4 line Y component.

IMGDMA1+0C54h Image Rotator 0 DMA FIFO Write Line Index Register IMGDMA1_IRT0_WRYIDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					YIDX											
Type					RO											

 © 2009 MediaTek Inc.

YIDX:

Display which FIFO line IRT0 DMA is now writing. Because IRT0 receives 4x4 block data, if its value is Y, then

IRT0 DMA is now writing (Y-3) th ~ Yth FIFO line. It will increase 4 while 4 line Y component are written into FIFO

completely.

When its value is equal to IRT0_FIFOYLEN (at the bottom of the FIFO) and current 4 line data are written into FIFO

completed, it will be reset to 4 to acknowledge IRT0 DMA to write to top of the FIFO again.

.

IMGDMA1+0C58h Image Rotator 0 DMA FIFO Read Line Index Register IMGDMA1_IRT0_RDYIDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**YIDX**											
Type					RO											

YIDX:

Display which FIFO line IRT0 DMA is now reading. If its value is Y, then IRT0 DMA is now reading Yth FIFO line.

When its value is equal to IRT0_FIFOYLEN (at the bottom of the FIFO segment) and current line data are read out, it

will be reset to 1 to acknowledge IRT0 DMA to read from top of the FIFO again.

IMGDMA1+0C80h Image Rotator 0 DMA Multiple Output Engine 1 Start Register IMGDMA1_IRT0_MO_1_STR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**STR**
Type																R/W
Reset																0

This register controls the activity of IRT0 Multiple Output 1. Note that before setting STR to “1”, all the configurations should be done by giving proper value. After setting the register, it would generate a starting pulse to set all controlling registers to initial values.

STR:

Start control for IRT0 Multiple Output 1

0: stop IRT0 Multiple Output 1

1: activate IRT0 Multiple Output 1

Note:

IRT0 Multiple Output module is used to transmit output pixel of either IRT0 DMA or VDODEC DMA with
Scan-Line mode to multiple destinations simultaneously, including MP4DEBLK, CRZ, PRZ, and IPP1. Due to its
destination located at Graphsys1 and Graphsys2 respectively, in order to transmit pixel data to different destinations
located at different Graphsys, IRT0 Multiple Output has been divided into IRT0 Multiple Output 0 and IRT0 Multiple
Output 1 to support multiple output function. ***The starting sequence of IRT0 DMA/VDODEC DMA with Scan-Line
mode, IRT0 Multiple Output 0 and IRT0 Multiple Output 1 should be as followed***.

IRT0 Multiple Output 0 started → IRT0 Multiple Output 1 started → IRT0 DMA / VDODEC DMA with
Scan-Line Mode

If IRT0 DMA / VDODEC DMA with Scan-Line Mode is allowed to transmit pixel data before IRT0 Multiple Output
0/1 are started, it would lose some pixels.

IMGDMA1+0C84h Image Rotator 0 DMA Multiple Output Engine 1 Control Register IMGDMA1_IRT0_MO_1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SRC						MP4DEBLK	PRZ	IMGDMA0
Type								R/W						R/W	R/W	R/W
Reset								0						0	0	0

IMGDMA0:

Enable output to Image DMA 0.

0: Disable

1: Enable

PRZ:

Enable output to PRZ.

0: Disable

1: Enable

MP4DEBLK:

Enable output to MP4DEBLK.

0: Disable

1: Enable

SRC:

Select input source of IRT0 Multiple Output 1.

0: IRT0 DMA

1: VDODEC scan-line engine

IMGDMA1+0C88h Image Rotator 0 DMA Multiple Output Engine 1 Busy Status IMGDMA1_IRT0_MO_1_BUSY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														MP4DEBLK BUSY	PRZ BUSY	IMGDMA0 BUSY
Type														RO	RO	RO

The BUSY bits are used to show if IRT0 Multiple Output 1 transmits data to destination or not. If there are two or more output path are enabled and only some BUSY bit stays at 1, then the IRT0 Multiple Output 1 may be stalled by the path.

IMGDMA0 BUSY:

0: No data transmit to Image DMA 0 now.

1: Data are transmitted to Image DMA 0 now.

PRZ BUSY:

0: No data transmit to PRZ now.

1: Data are transmitted to PRZ now.

MP4DEBLK BUSY:

0: No data transmit to MP4DEBLK now.

1: Data are transmitted to MP4DEBLK now.

5.16.3 Application Notes for Developing C Code

This section is for both hardware and software engineers when they develop their C codes. The codes in each following section can be copied into a C file if there are no other considerations. And the behavior of each module would be a little different if the values of #define are modified.

5.16.3.1 Image Buffer Write 1 DMA

```
#define IBW1_INT            0       // interrupt enabled
#define IBW1_SUCC           1       // auto restart
#define IBW1_TRIPLE         0       // triple buffers support under auto restart
#define IBW1_PITCH          0       // pitch function
#define IBW1_CLIP           0       // clip function
#define IBW1_LCD            1       // hardware trigger to LCD
#define IBW1_DC             0       // direct couple to LCD
#define IBW1_FORMAT         0       // frame buffer format, 0:RGB565, 1:RGB888, 2:ARGB8888
#define IBW1_ALPHA          128     // alpha value when writing ARGB8888 to memory
#define IBW1_SRC_XSIZE      100     // horizontal size of source image
#define IBW1_SRC_YSIZE      100     // vertical size of source image
```

 © 2009 MediaTek Inc.

```
#define IBW1_CLIP_L                11              // clip function horizontal starting pixel
#define IBW1_CLIP_R                90              // clip function horizontal ending pixel
#define IBW1_CLIP_T                21              // clip function vertical starting line
#define IBW1_CLIP_B                70              // clip function vertical ending line
#define IBW1_BASE_ADDR0            0x10100000      // base address of frame buffer 0
#define IBW1_BASE_ADDR1            0x10200000      // base address of frame buffer 1
#define IBW1_BASE_ADDR2            0x10300000      // base address of frame buffer 2
#define IBW1_BKGD_XSIZE0   150             // horizontal size of background image 0
#define IBW1_BKGD_XSIZE1   200             // horizontal size of background image 1
#define IBW1_BKGD_XSIZE2   250             // horizontal size of background image 2
        *IMGDMA0_IBW1_STR = 0x0;  // IBW1 DMA halts
        *IMGDMA0_IBW1_CON = (IBW1_FORMAT<<8) | (IBW1_DC<<7) | (IBW1_LCD<<6) |
                                        (IBW1_CLIP<<5)    |    (IBW1_PITCH<<4)    |
(IBW1_TRIPLE<<2) |
                                        (IBW1_SUCC<<1) | IBW1_INT;
        *IMGDMA0_IBW1_SRC_XSIZE = IBW1_SRC_XSIZE;
        *IMGDMA0_IBW1_SRC_YSIZE = IBW1_SRC_YSIZE;
        *IMGDMA0_IBW1_BASE_ADDR0 = IBW1_BASE_ADDR0;
        if (IBW1_FORMAT == 2)
                *IMGDMA0_IBW1_ALPHA = IBW1_ALPHA;
        if (IBW1_CLIP)
        {
                *IMGDMA0_IBW1_CLIPLR = (IBW1_CLIP_L<<16) + IBW1_CLIP_R;
                *IMGDMA0_IBW1_CLIPTB = (IBW1_CLIP_T<<16) + IBW1_CLIP_B;
        }
        if (IBW1_SUCC)
        {
                *IMGDMA0_IBW1_BASE_ADDR1 = IBW1_BASE_ADDR1;
                if (IBW1_TRIPLE)
                        *IMGDMA0_IBW1_BASE_ADDR2 = IBW1_BASE_ADDR2;
        }
        if (IBW1_PITCH)
        {
                *IMGDMA0_IBW1_BKGD_XSIZE0 = IBW1_BKGD_XSIZE0;
                if (IBW1_SUCC)
                {
                        *IMGDMA0_IBW1_BKGD_XSIZE1 = IBW1_BKGD_XSIZE1;
                        if (IBW1_TRIPLE)
                                *IMGDMA0_IBW1_BKGD_XSIZE2 = IBW1_BKGD_XSIZE2;
                }
        }
        *IMGDMA0_IBW1_STR = 0x1;  // IBW1 DMA starts
```

 © 2009 MediaTek Inc.

5.16.3.2 Image Buffer Write 2 DMA

```
#define IBW2_INT              0       // interrupt enabled
#define IBW2_LCD              0       // hardware trigger to LCD
#define IBW2_SUCC             0       // auto restart
#define IBW2_CLIP             0       // clip function
#define IBW2_OUT_DC           0       // direct couple to LCD
#define IBW2_OUT_IRT1         1       // output to IRT1
#define IBW2_OUT_R2Y0         0       // output to R2Y0
#define IBW2_ALPHA            128     // alpha value when direct couple to LCD
#define IBW2_XSIZE            100     // horizontal size of source image
#define IBW2_YSIZE            100     // vertical size of source image
#define IBW2_CLIP_L           11      // clip function horizontal starting pixel
#define IBW2_CLIP_R           90      // clip function horizontal ending pixel
#define IBW2_CLIP_T           21      // clip function vertical starting line
#define IBW2_CLIP_B           70      // clip function vertical ending line
#define IRT1_ULTRA_HIGH_EN    0       // IRT1 ultra high enable control
#define CAM_CRZ_VSYNC_EN0             // Camera frame synchronous reset signal enable control
        *IMGDMA0_IBW2_STR = 0x0;  // IBW2 DMA halts
        *IMGDMA0_IBW2_CON = (IBW2_OUT_R2Y0<<8) | (IBW2_OUT_IRT1<<7) | (IBW2_OUT_DC<<6) |
                            (IBW2_CLIP<<3) | (IBW2_SUCC<<2) | (IBW2_LCD<<1) | IBW2_INT |
                            (CAM_CRZ_VSYNC_EN<<5) | (IRT1_ULTRA_HIGH_EN<<4);
        if (IBW2_OUT_DC)
            *IMGDMA0_IBW2_ALPHA = IBW2_ALPHA;
        *IMGDMA0_IBW2_XSIZE = IBW2_XSIZE;
        *IMGDMA0_IBW2_YSIZE = IBW2_YSIZE;
        if (IBW2_CLIP)
        {
            *IMGDMA0_IBW2_CLIPLR = (IBW2_CLIP_L<<16) + IBW2_CLIP_R;
            *IMGDMA0_IBW2_CLIPTB = (IBW2_CLIP_T<<16) + IBW2_CLIP_B;
        }
        *IMGDMA0_IBW2_STR = 0x1;  // IBW2 DMA starts
```

5.16.3.3 Image Buffer Read 1 DMA

```
#define IBR1_INT              0       // interrupt enabled
#define IBR1_FORMAT           0       // pixel format of input frame buffer. 0:16-bits, 1: 24-bits
#define IBR1_ORDER            0       // pixel data order when pixel format is 24-bits. 0: BRG, 1: RGB
#define IBR1_BASE             0x10100000    // base address of input frame buffer
```

```
#define IBR1_PXLNUM                    10000   // pixel number of input frame buffer
        *IMGDMA0_IBR1_STR = 0x0;   // IBR1 DMA halts
        *IMGDMA0_IBR1_CON = (IBR1_ORDER<<2) | (IBR1_FORMAT<<1) | IBR1_INT;
        *IMGDMA0_IBR1_BASE = IBR1_BASE;
        *IMGDMA0_IBR1_PXLNUM = IBR1_PXLNUM;
        *IMGDMA0_IBR1_STR = 0x1;   // IBR1 DMA starts
```

5.16.3.4 Overlay DMA and OVL Multiple Output

```
#define OVL_MO_1_JPEG_DMA          1            // multiple output to JPEG DMA
#define OVL_MO_1_VDO_DMA1                   // multiple output to VDOENC DMA
#define OVL_MO_1_PRZ               1            // multiple output to PRZ
#define OVL_MO_0_IMGDMA1 1                  // multiple output to Image DMA 1
#define OVL_MO_0_Y2R0              1            // multiple output to Y2R0
#define OVL_MO_0_DRZ               1            // multiple output to DRZ
#define OVL_ENABLE                 0            // overlay function enabled
#define OVL_INT                         0           // interrupt enabled
#define OVL_MODE                   3            // mask data format
#define OVL_PALEN                  1            // palette read/write enable
#define OVL_PSEL                   1            // pixel engine selection
#define OVL_BASE                   0x10100000   // base address of mask
#define OVL_COLOR_KEY              0            // 8-bits color key
#define OVL_VRATIO                 2            // vertical scaling ratio
#define OVL_HARTIO                 2            // horizontal scaling ratio
#define OVL_XSIZE                  100          // vertical size of mask
#define OVL_YSIZE                  100          // horizontal size of mask
        *IMGDMA1_OVL_MO_1_STR = 0x0;    // OVL MO 1 halts
        *IMGDMA0_OVL_MO_0_STR = 0x0;    // OVL MO 0 halts
        *IMGDMA1_OVL_MO_1_CON = (OVL_MO_1_PRZ<<2) | (OVL_MO_1_VDO_DMA<<1) |
                                        OVL_MO_1_JPEG_DMA;
        *IMGDMA0_OVL_MO_0_CON = (OVL_MO_0_DRZ<<2) | (OVL_MO_0_Y2R0<<1) |
                                        OVL_MO_0_IMGDMA1;
        *IMGDMA1_OVL_MO_1_STR = 0x1;    // OVL MO 1 starts
        *IMGDMA0_OVL_MO_0_STR = 0x1;    // OVL MO 0 starts
        *IMGDMA0_OVL_STR = 0x0; // OVL DMA overlay function disable
        *IMGDMA0_OVL_CON = (OVL_PSEL<<5);
        if (OVL_ENABLE)
        {
                *IMGDMA0_OVL_CON = *IMGDMA0_OVL_CON | (OVL_PALEN<<4) | (OVL_MODE
<< 1) |
                                        OVL_INT;
                for(int  i=0;  i<256;  i++) // setting palette
                {
                        *(IMGDMA0_OVL_PAL_BASE + i) = (24-bits YUV color);
```

```
            }
            *IMGDMA0_OVL_BASE = OVL_BASE;
            *IMGDMA0_OVL_CFG = (OVL_COLOR_KEY<<8) | (OVL_VRATIO<<4) | OVL_HRATIO;
            *IMGDMA0_OVL_XSIZE = OVL_XSIZE;
            *IMGDMA0_OVL_YSIZE = OVL_YSIZE;
            *IMGDMA0_OVL_STR = 0x1; // OVL DMA overlay function enable
        }
```

5.16.3.5 Image Rotator 1 / Image Rotator 3 DMA

```
    // Note the following code could be applied to IRT3 DMA by replace IRT1 with IRT3
    #define IRT1_INT            1           // interrupt enabled
    #define IRT1_SUCC           0           // auto restart
    #define IRT1_TRIPLE         0           // triple buffers support under auto restart
    #define IRT1_LCD            1           // hardware trigger to LCD
    #define IRT1_PITCH          0           // pitch function
    #define IRT1_ROT            0           // rotation, 0: no rotation, 1: 90, 2: 180, 3: 270
    #define IRT1_FLIP           0           // flip after rotation
    #define IRT1_FORMAT         0           // frame buffer format, 0:RGB565, 1:RGB888, 2:ARGB8888
    #define IRT1_ALPHA          128         // alpha value when writing ARGB8888 to memory
    #define IRT1_SRC_XSIZE      100         // horizontal size of source image
    #define IRT1_SRC_YSIZE      100         // vertical size of source image
    #define IRT1_BASE_ADDR0     0x10100000  // base address of frame buffer 0
    #define IRT1_BASE_ADDR1     0x10200000  // base address of frame buffer 1
    #define IRT1_BASE_ADDR2     0x10300000  // base address of frame buffer 2
    #define IRT1_BKGD_XSIZE0    150         // horizontal size of background image 0
    #define IRT1_BKGD_XSIZE1    200         // horizontal size of background image 1
    #define IRT1_BKGD_XSIZE2    250         // horizontal size of background image 2
    #define IRT1_FIFO_BASE      0x40000000  // FIFO base address when rotated 90 or 270
        *IMGDMA0_IRT1_STR = 0x0;    // IRT1 DMA halts
        *IMGDMA0_IRT1_CON = (IRT1_FORMAT<<8) | (IRT1_FLIP<<7) | (IRT1_ROT<<5) |
                            (IRT1_PITCH<<4) | (IRT1_LCD<<3) | (IRT1_TRIPLE<<2) |
                            (IRT1_SUCC<<1) | IRT1_INT;
        *IMGDMA0_IRT1_SRC_XSIZE = IRT1_SRC_XSIZE;
        *IMGDMA0_IRT1_SRC_YSIZE = IRT1_SRC_YSIZE;
        *IMGDMA0_IRT1_BASE_ADDR0 = IRT1_BASE_ADDR0;
        if (IRT1_ROT == 1 || IRT1_ROT == 3)
            *IMGDMA0_IRT1_FIFO_BASE = IRT1_FIFO_BASE;
```

```
if (IRT1_FORMAT == 2)
        *IMGDMA0_IRT1_ALPHA = IRT1_ALPHA;
if (IRT1_SUCC)
{
        *IMGDMA0_IRT1_BASE_ADDR1 = IRT1_BASE_ADDR1;
        if (IRT1_TRIPLE)
                *IMGDMA0_IRT1_BASE_ADDR2 = IRT1_BASE_ADDR2;
}
if (IRT1_PITCH)
{
        *IMGDMA0_IRT1_BKGD_XSIZE0 = IRT1_BKGD_XSIZE0;
        if (IRT1_SUCC)
        {
                *IMGDMA0_IRT1_BKGD_XSIZE1 = IRT1_BKGD_XSIZE1;
                if (IRT1_TRIPLE)
                        *IMGDMA0_IRT1_BKGD_XSIZE2 = IRT1_BKGD_XSIZE2;
        }
}
*IMGDMA0_IRT1_STR = 0x1;   // IRT1 DMA starts
```

5.16.3.6 JPEG Encoder DMA

```
#define JPEG_INT                0       // interrupt enabled
#define JPEG_MODE               0       // working mode. 0: YUV422, 1: gray, 2: YUV420, 3: YUV411
#define JPEG_SUCC               0       // auto restart
#define JPEG_FIFO_BASE          0x40000000      // FIFO base address
#define JPEG_FIFOLEN            16      // 16 for double buffers
#define JPEG_XSIZE              800     // horizontal size of input image
#define JPEG_YSIZE              640     // vertical size of input image
#define CAM_CRZ_VSYNC_EN0               // Camera frame synchronous reset signal enable control
        *IMGDMA1_JPEG_STR = 0x0; // JPEG DMA halts
        *IMGDMA1_JPEG_CON = (JPEG_SUCC<<3) | (JPEG_MODE<<1) | JPEG_INT |
                                        (CAM_CRZ_VSYNC_EN<<7);
        *IMGDMA1_JPEG_FIFO_BASE = JPEG_FIFO_BASE;
        *IMGDMA1_JPEG_FIFOLEN = JPEG_FIFOLEN;
        *IMGDMA1_JPEG_XSIZE = JPEG_XSIZE;
        *IMGDMA1_JPEG_YSIZE = JPEG_YSIZE;
        *IMGDMA1_JPEG_STR = 0x1; // JPEG DMA starts
```

5.16.3.7 Video Encode DMA

```
#define VDOENC_WR_INT           1       // frame buffer write engine interrupt enabled
#define VDOENC_W2R              1       // trigger of read engine from write engine
```


© 2009 MediaTek Inc.

```
        #define VDOENC_SUCC                 1                       // auto restart
        #define VDOENC_RD_INT               0                       // frame buffer read engine interrupt enabled
        #define VDOENC_ROT                  0                       // rotation of read engine, 0: no rotation, 1:
90, 2: 180, 3: 270
        #define VDOENC_FLIP                 0                       // flip after rotation
        #define VDOENC_XSIZE                176                     // horizontal size of input image
        #define VDOENC_YSIZE                144                     // vertical size of input image
        #define VDOENC_PXLNUM               25344   // pixel number of input image
        #define VDOENC_Y_BASE1              0x10000000              // Y base address of frame buffer 1
        #define VDOENC_U_BASE1              0x10006300              // U base address of frame buffer 1
        #define VDOENC_V_BASE1              0x10007BC0              // V base address of frame buffer 1
        #define VDOENC_Y_BASE2              0x10100000              // Y base address of frame buffer 2
        #define VDOENC_U_BASE2              0x10106300              // U base address of frame buffer 2
        #define VDOENC_V_BASE2              0x10107BC0              // V base address of frame buffer 2
        #define VDOENC_LB_BASE              0x40000000              // base address of line buffer
        #define VDOENC_LB_YLEN                                      8       // 8 for double buffers
        #define VDOENC_ULTRA_HIGH_EN                0               // 8 for double buffers
                *IMGDMA1_VDOENC_STR = 0x0; // VDOENC DMA halts
                *IMGDMA1_VDOENC_CON = (VDOENC_FLIP<<6) | (VDOENC_ROT<<4) |
(VDOENC_RD_INT<<3) |
                                                            (VDOENC_SUCC<<2) |
(VDOENC_W2R<<1) | VDOENC_WR_INT |
                                                            (VDOENC_ULTRA_HIGH_EN<<7);
                *IMGDMA1_VDOENC_Y_BASE1 = VDOENC_Y_BASE1;
                *IMGDMA1_VDOENC_U_BASE1 = VDOENC_U_BASE1;
                *IMGDMA1_VDOENC_V_BASE1 = VDOENC_V_BASE1;
                if (VDOENC_SUCC)
                {
                        *IMGDMA1_VDOENC_Y_BASE2 = VDOENC_Y_BASE2;
                        *IMGDMA1_VDOENC_U_BASE2 = VDOENC_U_BASE2;
                        *IMGDMA1_VDOENC_V_BASE2 = VDOENC_V_BASE2;
                }
                *IMGDMA1_VDOENC_XSIZE = VDOENC_XSIZE;
                *IMGDMA1_VDOENC_YSIZE = VDOENC_YSIZE;
                *IMGDMA1_VDOENC_PXLNUM = VDOENC_PXLNUM;
                *IMGDMA1_VDOENC_LB_BASE = VDOENC_LB_BASE;
                *IMGDMA1_VDOENC_LB_YLEN = VDOENC_LB_YLEN;
                *IMGDMA1_VDOENC_STR = 0x1; // VDOENC DMA starts
```

5.16.3.8 Video Decode DMA

```
        #define VDODEC_INT                  0                       // interrupt enabled
        #define VDODEC_ROT                  0                       // rotation of read engine, 0: no rotation, 1:
90, 2: 180, 3: 270
```

 © 2009 MediaTek Inc.

```
#define VDODEC_FLIP            0            // flip after rotation
#define VDODEC_SCAN_MODE       0            // frame buffer mode selection, 0: 4x4 block, 1: scan line
#define VDODEC_XSIZE           176          // horizontal size of input image
#define VDODEC_YSIZE           144          // vertical size of input image
#define VDODEC_PXLNUM          25344  // pixel number of input image
#define VDODEC_Y_BASE          0x10000000   // Y base address of frame buffer
#define VDODEC_U_BASE          0x10006300   // U base address of frame buffer
#define VDODEC_V_BASE          0x10007BC0   // V base address of frame buffer
        *IMGDMA1_VDODEC_STR = 0x0; // VDODEC DMA halts
        *IMGDMA1_VDODEC_CON = (VDODEC_SCAN_MODE<<7) | (VDODEC_FLIP<<6) |
                              (VDODEC_ROT<<4) | VDODEC_INT;
        *IMGDMA1_VDODEC_Y_BASE = VDODEC_Y_BASE;
        *IMGDMA1_VDODEC_U_BASE = VDODEC_U_BASE;
        *IMGDMA1_VDODEC_V_BASE = VDODEC_V_BASE;
        *IMGDMA1_VDODEC_XSIZE = VDODEC_XSIZE;
        *IMGDMA1_VDODEC_YSIZE = VDODEC_YSIZE;
        *IMGDMA1_VDODEC_PXLNUM = VDODEC_PXLNUM;
        *IMGDMA1_VDODEC_STR = 0x1; // VDODEC DMA starts
```

5.16.3.9 Image Rotator 0 DMA and IRT0 Multiple Output

```
#define IRT0_MO_0_CRZ          0            // multiple output to CRZ
#define IRT0_MO_0_IPP1         0            // multiple output to IPP1
#define IRT0_MO_1_SRC          0            // source selection, 0: IRT0 DMA, 1: VDODEC scan engine
#define IRT0_MO_1_MP4DEBLK     0            // multiple output to MP4DEBLK
#define IRT0_MO_1_PRZ          1            // multiple output to PRZ
#define IRT0_MO_1_IMGDMA0 0        // multiple output to Image DMA 0
#define IRT0_INT               0            // interrupt enabled
#define IRT0_SUCC              0            // auto restart
#define IRT0_SRC               0            // source selection, 0: VDOENC RDMA, 1: VDODEC DMA
#define IRT0_ROT               0            // rotation of read engine, 0: no rotation, 1: 90, 2: 180, 3: 270
#define IRT0_FLIP              0            // flip after rotation
#define IRT0_FIFO_BASE         0x40000000   // FIFO base address
#define IRT0_FIFOYLEN          16           // 16 for double buffers
#define IRT0_XSIZE             176          // horizontal size of video frame buffer
#define IRT0_YSIZE             144          // vertical size of video frame buffer
        *IMGDMA0_IRT0_MO_0_STR = 0x0;   // IRT0 MO 0 halts
        *IMGDMA1_IRT0_MO_1_STR = 0x0;   // IRT0 MO 1 halts
        *IMGDMA0_IRT0_MO_0_CON = (IRT0_MO_0_IPP1<<1) | IRT0_MO_0_CRZ;
        *IMGDMA1_IRT0_MO_1_CON = (IRT0_MO_1_SRC<<8) | (IRT0_MO_1_MP4DEBLK<<2) |
```

(IRT0_MO_1_PRZ<<1) |
IRT0_MO_1_IMGDMA0;
IMGDMA0_IRT0_MO_0_STR = 0x1; // IRT0 MO 0 starts
IMGDMA1_IRT0_MO_1_STR = 0x1; // IRT0 MO 1 starts
if (VDODEC_SCAN_MODE == 0) // definition of VDODEC_SCAN_MODE, refer to previous section
{
IMGDMA1_IRT0_STR = 0x0; // IRT0 DMA halts
*IMGDMA1_IRT0_CON = (IRT0_FLIP<<6) | (IRT0_ROT<<4) | (IRT0_SRC<<2) | (IRT0_SUCC<<1) |
IRT0_INT;
*IMGDMA1_IRT0_FIFO_BASE = IRT0_FIFO_BASE;
*IMGDMA1_IRT0_FIFOYLEN = IRT0_FIFOYLEN;
*IMGDMA1_IRT0_XSIZE = IRT0_XSIZE;
*IMGDMA1_IRT0_YSIZE = IRT0_YSIZE;
IMGDMA1_IRT0_STR = 0x1; // IRT0 DMA starts
}

5.16.4 A Quick Guideline for Driver Developer from TK6516 to MT6516

In MT6516, most modules are inherited from TK6516. Therefore, most register settings are same between the two version. Here we point out the differences when immigrating driver from TK6516 to MT6516.

5.16.4.1 Global View

There are two extreme modifications when immigrating from TK6516 to MT6516.

1. The first is **Image DMA is divided into Image DMA 0 and Image DMA 1**. In Image DMA 1, it contains video and jpeg related modules, including VDOENC DMA, VDODEC DMA, IRT0 DMA, and JPEG DMA.
The remaining modules are placed in Image DMA 0. Due to the modification, the prefix of each register has been modified from IMGDMA_ to either IMGDMA0_ or IMGDMA1_.
2. The second extreme modification is IBR2 and IRT2 DMA have been removed from Image DMA in MT6516.
All the settings to these two modules are invalid now.

Besides, **the address setting of each sub-module should be changed from 4x to 8x address, except base address of IRT1 and IRT3**. The detail address limit could refer to register definitions in the previous section.

5.16.4.2 Local View

The modifications of registers of each sub-module are described as follows.

5.16.4.2.1 IBW1 DMA

Register removed:

IMGDMA_IBW1_DPITCH1

IMGDMA_IBW1_DPITCH2

Register added:

IMGDMA0_IBW1_BASE_ADDR2 ➔ base address of the third frame buffer
IMGDMA0_IBW1_BKGD_XSIZE0 ➔ horizontal size of the first background image
IMGDMA0_IBW1_BKGD_XSIZE1 ➔ horizontal size of the second background image
IMGDMA0_IBW1_BKGD_XSIZE2 ➔ horizontal size of the third background image

Register modified:

IMGDMA_IBW1_BSADDR1 ➔ IMGDMA0_IBW1_BAES_ADDR0
IMGDMA_IBW1_BSADDR2 ➔ IMGDMA0_IBW1_BAES_ADDR1
IMGDMA_IBW1_HSIZE (set n-1) ➔ IMGDMA0_IBW1_SRC_XSIZE (set n)
IMGDMA_IBW1_VSIZE (set n-1) ➔ IMGDMA0_IBW1_SRC_YSIZE (set n)
IMGDMA_IBW1_CLIPLR (set n-1) ➔ IMGDMA0_IBW1_CLIP_LR (set n)
IMGDMA_IBW1_CLIPTB (set n-1) ➔ IMGDMA0_IBW1_CLIP_TB (set n)
IMGDMA_IBW1_CON ➔ IMGMDA0_IBW1_CON

MT6516 IMGDMA0_IBW1_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							FMT		DC	LCD	CLIP	PITCH		TRIPLE	AUTO RSTR	IT
Type							R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset							0		0	0	0	0		0	0	0

TK6516 IMGDMA_IBW1_CON

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FMT		CLIP	DC	AUTO RSTR	LCD	PITCH	IT
Type									R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset									0		0	0	0	0	0	0

5.16.4.2.2 IBW2 DMA

Register modified:

IMGDMA_IBW2_HSIZE (set n-1) ➔ IMGDMA0_IBW2_XSIZE (set n)
IMGDMA_IBW2_VSIZE (set n-1) ➔ IMGDMA0_IBW2_YSIZE (set n)
IMGDMA_IBW2_CLIPLR (set n-1) ➔ IMGDMA0_IBW2_CLIP_LR (set n)
IMGDMA_IBW2_CLIPTB (set n-1) ➔ IMGDMA0_IBW2_CLIP_TB (set n)
IMGDMA_IBW2_CON ➔ IMGDMA0_IBW2_CON

MT6516 IMGDMA0_IBW2_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								R2Y0	IRT1	DC	CAM_CRZ_VSYNC_EN	IRT1_ULTRA_HIGH_EN	CLIP	AUTO RSTR	LCD	IT
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

TK6516 IMGDMA_IBW2_CON

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									IRT1	DC			CLIP	AUTO RSTR	LCD	IT

Type									R/W	R/W			R/W	R/W	R/W	R/W
Reset									0	0			0	0	0	0

5.16.4.2.3 Image Buffer Read 1 DMA

Register modified:

IMGDMA_IBR1_PXLNUM (set n-1) ➔ IMGDMA0_IBR1_PXLNUM (set n)

5.16.4.2.4 OVL DMA and OVL Multiple Output

Register added:

IMGDMA0_OVL_MO_0_STR
IMGDMA0_OVL_MO_0_CON
IMGDMA1_OVL_MO_1_STR
IMGDMA1_OVL_MO_1_CON

Register modified:

IMGDMA_OVL_BSADDR ➔ IMGDMA0_OLV_BASE
IMGDMA_OVL_HSIZE ➔ IMGDMA0_OVL_XSIZE
IMGDMA_OVL_VSIZE ➔ IMGDMA0_OVL_YSIZE
IMGDMA_OVL_CON ➔ IMGDMA0_OVL_CON

MT6516 IMGDMA0_OVL_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**PSEL**	**PALEN**		**MODE 1**	**MODE 0**	**IT**
Type											R/W	R/W		R/W	R/W	R/W
Reset											0	0		0	0	0

TK6516 IMGDMA_OVL_CON

Name						**PRZ**	**DRZ**	**Y2R0**	**JPEG**	**VDOE NC**	**PSEL**	**PALEN**	**AUTO RSTR**	**MODE 1**	**MODE 0**	**IT**
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0

5.16.4.2.5 IRT1 / IRT3 DMA

Register removed:

IMGDMA_IRT1_LINE_DPITCH1
IMGDMA_IRT1_LINE_DPITCH2
IMGDMA_IRT1_FRM_DPITCH1
IM GDMA_IRT1_FRM_DPITCH2
IMGDMA_IRT1_FIFOYLEN

Register added:

IMGDMA0_IRT1_BASE_ADDR2 ➔ base address of the third frame buffer
IMGDMA0_IRT1_BKGD_XSIZE0 ➔ horizontal size of the first background image
IMGDMA0_IRT1_BKGD_XSIZE1 ➔ horizontal size of the second background image
IMGDMA0_IRT1_BKGD_XSIZE2 ➔ horizontal size of the third background image

Register modified:

IMGDMA_IRT1_BSADDR1 ➔ IMGDMA0_IRT1_BAES_ADDR0
IMGDMA_IRT1_BSADDR2 ➔ IMGDMA0_IRT1_BAES_ADDR1

IMGDMA_IRT1_HSIZE (set n-1) ➔ IMGDMA0_IRT1_SRC_XSIZE (set n)
IMGDMA_IRT1_VSIZE (set n-1) ➔ IMGDMA0_IRT1_SRC_YSIZE (set n)
IMGDMA_IRT1_CON ➔ IMGMDA0_IRT1_CON

MT6516 IMGDMA0_IRT1_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						BUFFER DOUBLE	FMT		FLIP	ROT		PITCH	LCD	TRIPLE	AUTO RSTR	IT
Type							R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset							0		0	0		0	0	0	0	0

TK6516 IMGDMA_IRT1_CON

Name						LCD	PITCH	OUT FMT				FLIP	ROT		AUTO RSTR	IT
Type						R/W	R/W	R/W				R/W	R/W		R/W	R/W
Reset						0	0	0				0	0		0	0

5.16.4.2.6 JPEG DMA

Register modified:

IMGDMA_JPEG_BSADDR ➔ IMGMDA1_JPEG_FIFO_BASE
IMGDMA_JPEG_HSIZE (set n-1) ➔ IMGDMA1_JPEG_XSIZE (set n)
IMGMDA_JPEG_VSIZE (set n-1) ➔ IMGDMA1_JPEG_YSIZE (set n)

5.16.4.2.7 VDOENC DMA

Register added:

IMGDMA1_VDOENC_LB_BASE
IMGDMA1_VDOENC_LB_YLEN

Register modified:

IMGDMA_VDOENC_HSIZE ➔ IMGDMA1_VDOENC_XSIZE
IMGDMA_VDOENC_VSIZE ➔ IMGDMA1_VDOENC_YSIZE
IMGDMA_VDOENC_CON ➔ IMGDMA1_VDOENC_CON

MT6516 IMGDMA1_VDOENC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									VDONEC_ULTRA_HIGH_EN	FLIP	ROT		RD IT	AUTO RSTR	W2R	WR IT
Type									R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset									0	0	0		0	0	0	0

TK6516 IMGDMA_VDOENC_CON

Name									BYPS	FLIP	ROT		RD IT	AUTO RSTR	W2R	WR IT
Type									R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset									0	0	0		0	0	0	0

© 2009 MediaTek Inc.

5.16.4.2.8 VDODEC DMA

Register modified:

IMGDMA_VDODEC_HSIZE ➔ IMGDMA1_VDODEC_XSIZE
IMGDMA_VDODEC_VSIZE ➔ IMGDMA1_VDODEC_YSIZE
IMGDMA_VDODEC_CON ➔ IMGDMA1_VDODEC_CON

MT6516 IMGDMA1_VDODEC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SCAN MODE**	**FLIP**	**ROTATION**					**IT**
Type									R/W	R/W	R/W					R/W
Reset									0	0	0					0

TK6516 IMGDMA_VDODEC_CON

Name									**BYPS**	**FLIP**	**ROTATION**		**H264**			**DONE IT**
Type									R/W	R/W	R/W		R/W			R/W
Reset									0	0	0		0			0

5.16.4.2.9 IRT0 DMA and IRT0 Multiple Output

Register removed:

IMGDMA_IRT0_PXLNUM

Register added:

IMGDMA0_IRT0_MO_0_STR
IMGDMA0_IRT0_MO_0_CON
IMGDMA1_IRT0_MO_1_STR
IMGDMA1_IRT0_MO_1_CON

Register modified:

IMGDMA_IRT0_BASE ➔ IMGDMA1_IRT0_FIFO_BASE
IMGDMA_IRT0_HSIZE ➔ IMGDMA1_IRT0_XSIZE
IMGDMA_IRT0_VSIZE ➔ IMGDMA1_IRT0_YSIZE
IMGDMA_IRT0_CON ➔ IMGDMA1_IRT0_CON

MT6516 IMGDMA1_IRT0_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**FLIP**	**ROT**			**SRCSEL**	**AUTO RSTR**	**IT**
Type										R/W	R/W			R/W	R/W	R/W
Reset										0	0			0	0	0

TK6516 IMGDMA_IRT0_CON

Name					**MP4D EBLK**	**IPP1**	**PRZ**	**CRZ**	**BYPS**	**FLIP**	**ROT**		**H264**	**PSEL**	**AUTO RSTR**	**IT**
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0		0	0	0	0

© 2009 MediaTek Inc.

5.17 Image Processor

The Image processing processor consists of the three functional blocks shown in Figure 137. The first functional block consists of IPP1, Y2R0, and IPP2. The second one consists of R2Y0, and the last one consists of Y2R1. For each functional block, only one input source can be active, and only one output destination can be active, too. The input of the IPP1 can be only in YUV format. The input of the R2Y0, of course, can be only in RGB format. The input of the Y2R1 can be only in YUV format.

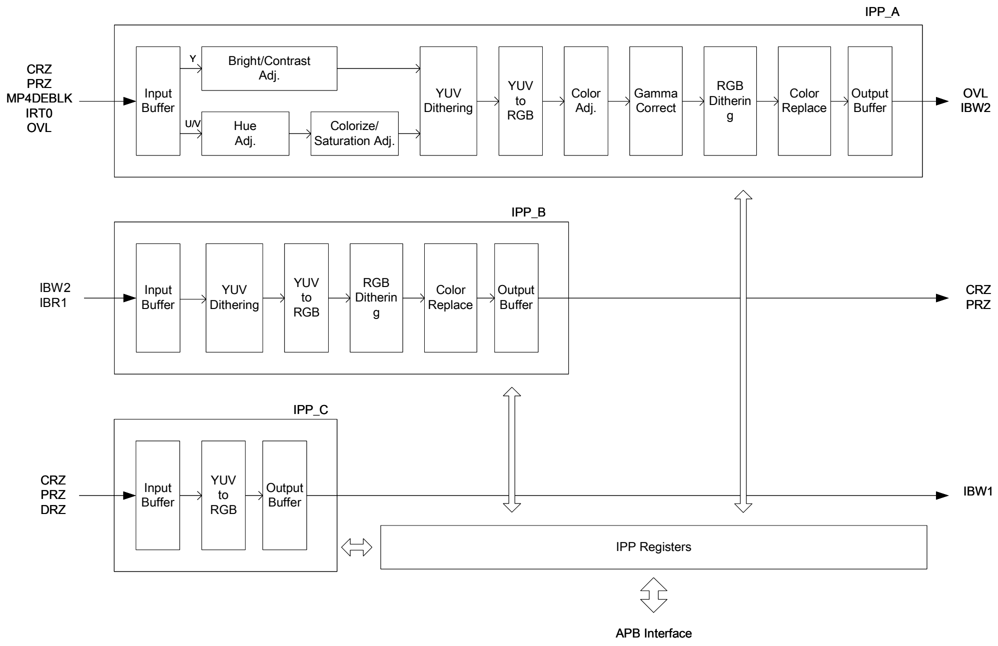


Figure 137: Image processing processor (IPP) block diagram.

5.17.1 Register Definitions

Register Address	Register Function	Acronym
IMG+0000h	IMGPROC IPP Configuration register	**IMGPROC_IPP_CFG**
IMG+0004h	IMGPROC R2Y0 Configuration register	**IMGPROC_R2Y0_CFG**
IMG+0008h	IMGPROC Y2R1 Configuration register	**IMGPROC_Y2R1_CFG**
IMG+000Ch	IPP SDT Control register	**IMGPROC_IPP_SDTCON**
IMG+0010h	Y2R1 SDT Control register	**IMGPROC_Y2R1_SDTCON**
IMG+0100h	Hue adjustment coefficient C11	**IMGPROC_HUE11**
IMG+0104h	Hue adjustment coefficient C12	**IMGPROC_HUE12**
IMG+0108h	Hue adjustment coefficient C21	**IMGPROC_HUE21**
IMG+010Ch	Hue adjustment coefficient C22	**IMGPROC_HUE22**
IMG+0110h	Saturation adjustment coefficient	**IMGPROC_SAT**
IMG+0120h	Brightness adjustment coefficient B1	**IMGPROC_BRIADJ1**
IMG+0124h	Brightness adjustment coefficient B2	**IMGPROC_BRIADJ2**

IMG+0128h	Contrast adjustment coefficient	**IMGPROC_CONADJ**
IMG+0130h	Colorize effect coefficient	**IMGPROC_COLORIZEU**
IMG+0134h	Colorize effect coefficient	**IMGPROC_COLORIZEV**
IMG+0170h	Gamma correction offset for segment 0	**IMGPROC_GAMMA_OFF0**
IMG+0174h	Gamma correction offset for segment 1	**IMGPROC_GAMMA_OFF1**
IMG+0178h	Gamma correction offset for segment 2	**IMGPROC_GAMMA_OFF2**
IMG+017Ch	Gamma correction offset for segment 3	**IMGPROC_GAMMA_OFF3**
IMG+0180h	Gamma correction offset for segment 4	**IMGPROC_GAMMA_OFF4**
IMG+0184h	Gamma correction offset for segment 5	**IMGPROC_GAMMA_OFF5**
IMG+0188h	Gamma correction offset for segment 6	**IMGPROC_GAMMA_OFF6**
IMG+018Ch	Gamma correction offset for segment 7	**IMGPROC_GAMMA_OFF7**
IMG+0190h	Gamma correction slope for segment 0	**IMGPROC_GAMMA_SLP0**
IMG+0194h	Gamma correction slope for segment 1	**IMGPROC_GAMMA_SLP1**
IMG+0198h	Gamma correction slope for segment 2	**IMGPROC_GAMMA_SLP2**
IMG+019Ch	Gamma correction slope for segment 3	**IMGPROC_GAMMA_SLP3**
IMG+01A0h	Gamma correction slope for segment 4	**IMGPROC_GAMMA_SLP4**
IMG+01A4h	Gamma correction slope for segment 5	**IMGPROC_GAMMA_SLP5**
IMG+01A8h	Gamma correction slope for segment 6	**IMGPROC_GAMMA_SLP6**
IMG+01ACh	Gamma correction slope for segment 7	**IMGPROC_GAMMA_SLP7**
IMG+01B0h	Gamma correction control register	**IMGPROC_GAMMA_CON**
IMG+0200h	Color adjustment offset x for red segment 1	**IMGPROC_COLOR1R_OFFX**
IMG+0204h	Color adjustment offset x for red segment 2	**IMGPROC_COLOR2R_OFFX**
IMG+0208h	Color adjustment offset x for green segment 1	**IMGPROC_COLOR1G_OFFX**
IMG+020Ch	Color adjustment offset x for green segment 2	**IMGPROC_COLOR2G_OFFX**
IMG+0210h	Color adjustment offset x for blue segment 1	**IMGPROC_COLOR1B_OFFX**
IMG+0214h	Color adjustment offset x for blue segment 2	**IMGPROC_COLOR2B_OFFX**
IMG+0220h	Color adjustment offset y for red segment 1	**IMGPROC_COLOR1R_OFFY**
IMG+0224h	Color adjustment offset y for red segment 2	**IMGPROC_COLOR2R_OFFY**
IMG+0228h	Color adjustment offset y for green segment 1	**IMGPROC_COLOR1G_OFFY**
IMG+022Ch	Color adjustment offset y for green segment 2	**IMGPROC_COLOR2G_OFFY**
IMG+0230h	Color adjustment offset y for blue segment 1	**IMGPROC_COLOR1B_OFFY**
IMG+0234h	Color adjustment offset y for blue segment 2	**IMGPROC_COLOR2B_OFFY**
IMG+0240h	Color adjustment slope for red segment 0	**IMGPROC_COLOR1G_SLP**
IMG+0244h	Color adjustment slope for red segment 1	**IMGPROC_COLOR1G_SLP**
IMG+0248h	Color adjustment slope for red segment 2	**IMGPROC_COLOR2G_SLP**
IMG+0250h	Color adjustment slope for red segment 0	**IMGPROC_COLOR1G_SLP**
IMG+0254h	Color adjustment slope for red segment 1	**IMGPROC_COLOR1G_SLP**
IMG+0258h	Color adjustment slope for red segment 2	**IMGPROC_COLOR2G_SLP**

IMG+0260h	Color adjustment slope for red segment 0	**IMGPROC_COLOR1G_SLP**
IMG+0264h	Color adjustment slope for red segment 1	**IMGPROC_COLOR1G_SLP**
IMG+0268h	Color adjustment slope for red segment 2	**IMGPROC_COLOR2G_SLP**
IMG+0318h	Input source and output destination selection	**IMGPROC_IO_MUX**
IMG+0320h	Image engine process enable	**IMGPROC_EN**
IMG+0324h	IPP RGB value detect	**IMGPROC_IPP_RGB_DETECT**
IMG+0328h	IPP RGB value replace	**IMGPROC_IPP_RGB_REPLACE**
IMG+032Ch	Y2R1 RGB value detect	**IMGPROC_Y2R1_RGB_DETECT**
IMG+0330h	Y2R1 RGB value replace	**IMGPROC_Y2R1_RGB_REPLACE**

Table 108 ***Image Engine Registers***

IMG+0000h IMGPROC IPP Configuration register IMGPROC_IPP_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		IPP_RGB_DETECT_EN	SDT1_EN	INV_EN				GAMMA_EN				CLRADJ_EN			ROUND_Y2R0	Y2R0_EN
Type		R/W	R/W	R/W				R/W				R/W			R/W	R/W
Reset		0	0	0				0				0			0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				SDT0_EN			CLRIZE_EN	SAT_EN			HUE_EN	CB_EN				
Type				R/W			R/W	R/W			R/W	R/W				
Reset				0			0	0			0	0				

IPP_RGB_DETECT_EN: The enable bit of the “RGB detect and replace” process
SDT1_EN: The enable bit of spatial dithering process for SDT1 (RGB Domain)
INV_EN: The enable bit of color inversion
GAMMA_EN: The enable bit of gamma correction
CLRADJ_EN: The enable bit of color adjustment
ROUND_Y2R0: The rounding control bit of yuv2rgb process
Y2R0_EN: The enable bit of YUV2RGB conversion
SDT0_EN: The enable bit of spatial dithering process for SDT0 (YUV Domain)
CLRIZE_EN: The enable bit of colorize
SAT_EN: The enable bit of saturation adjustment
HUE_EN: The enable bit of hue adjustment
CB_EN: The enable bit of contrast and brightness adjustment

NOTE: SDT0_EN and SDT1_EN can NOT be ‘1’ simultaneously.

IMG+0004h IMGPROC R2Y0 Configuration register IMGPROC_R2Y0_CFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															ROUND_R2Y0	R2Y0_EN
Type															RW	RW
Reset															0	0

R2Y0_EN: The enable bit of RGB2YUV conversion in R2Y0
ROUND_R2Y0: The rounding control bit of conversion process

IMG+0008h IMGPROC Y2R1 Configuration register IMGPROC_Y2R1_CFG

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**Y2R1_RGB_DETECT_EN**	**SDT3_EN**	**SDT2_EN**	**ROUND_Y2R1**	**Y2R1_EN**
Type												RW	RW	RW	RW	RW
Reset												0	0	0	0	0

Y2R1_RGB_DETECT_EN: The enable bit of "RGB detect and replace" process
SDT3_EN: The enable bit of spatial dithering process in RGB domain
SDT2_EN: The enable bit of spatial dithering process in YUV domain
ROUND_Y2R1: The rounding control bit of conversion process
Y2R1_EN: The enable bit of YUV2RGB conversion in Y2R1
NOTE: SDT2_EN and SDT3_EN can NOT be '1' simultaneously.

IMG+000Ch IPP SDT Control Register IMGPROC_IPP_SDTCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**IPP_SEED1**				**IPP_SEED2**			
Type									R/W				R/W			
Reset									0				0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IPP_SEED3**					**IPP_BD1**				**IPP_BD2**				**IPP_BD3**		
Type	R/W					R/W				R/W				R/W		
Reset	0					0				0				0		

IPP_BD1-3: The registers are used to configure the bit number of dithering. (e.g. 888 to 565, BD1=3;BD2=2;BD3=3)
IPP_SEED1-3: The registers are used to configure the seeds (LSB 4-bit) of the three pseudo-random code generators.
NOTE: IPP_SEED1-3 should be configured when the enable bit (+0x0320) is '0'.

IMG+0010h Y2R1 SDT Control Register IMGPROC_Y2R1_SDTCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**Y2R1_SEED1**				**Y2R1_SEED2**			
Type									R/W				R/W			
Reset									0				0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y2R1_SEED3**					**Y2R1_BD1**				**Y2R1_BD2**				**Y2R1_BD3**		
Type	R/W					R/W				R/W				R/W		
Reset	0					0				0				0		

IPP_BD1-3: The registers are used to configure the bit number of dithering. (e.g. 888 to 565, BD1=3;BD2=2;BD3=3)
IPP_SEED1-3: The registers are used to configure the seeds (LSB 4-bit) of the three pseudo-random code generators.

IMG+0100h Hue adjustment coefficient C11 IMGPROC_HUE11

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**C11**							
Type									R/W							
Reset									40h							

IMG+0104h Hue adjustment coefficient C12 IMGPROC_HUE12

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**C12**							
Type									R/W							
Reset									0							

IMG+0108h Hue adjustment coefficient C21 IMGPROC_HUE21

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**C21**							
Type									R/W							
Reset									0							

IMG+010Ch Hue adjustment coefficient C22 IMGPROC_HUE22

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**C22**							
Type									R/W							
Reset									40h							

This registers control the parameters of hue adjustment for the image. The effect is performed on the U and V component in YUV color space. The user should specify the coefficients that form the transformation matrix. The formula is listed as follows:

$$\begin{bmatrix} u_0 \\ v_o \end{bmatrix} = \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix} \cdot \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

$$where \quad C11 = 64\cos\theta, \quad C12 = 64\sin\theta, \quad C21 = -64\sin\theta, \quad C22 = 64\cos\theta$$

The coefficients are in 2's complement format and range from C0h to 40h (from –64 to 64 in decimal, while 64 is normalized to 1). Any value beyond this range is invalid.

For example, to rotate the color space counterclockwise by 30 degree, the coefficients should be 37h, 20h, e0h, and 37h.

C11: C11 of the transformation matrix in 2's complement format

C12: C12 of the transformation matrix in 2's complement format

C21: C21 of the transformation matrix in 2's complement format

C22: C22 of the transformation matrix in 2's complement format

 © 2009 MediaTek Inc.

IMG+0110h Saturation adjustment coefficient IMGPROC_SATADJ

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**SAT**						
Type										R/W						
Reset										20h						

This register defines the parameter of saturation adjustment for the image. The basics of saturation tuning is to multiply the U and V component by a scaling factor, which could range from 0 to 127, to degrade or enhance the strength on color components. Setting to 20h represents no scaling.

SAT: Saturation coefficient.

IMG+0120h Brightness adjustment coefficient B1 IMGPROC_BRIADJ1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BRI**							
Type									R/W							
Reset									0							

This register defines the parameter of brightness adjustment for the image. The parameter is in unsigned format. Setting the value to be greater than 0 adds to the intensity of the image pixel. In terms of transfer curve, it represents the offset in the y-axis. The valid value ranges from 0 to 255.

BRI: Brightness adjustment coefficient.

IMG+0124h Brightness adjustment coefficient B2 IMGPROC_BRIADJ2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DRK**							
Type									R/W							
Reset									0							

This register controls the parameter of brightness adjustment for the image. The parameter is in unsigned format. Setting the value to be greater than 0 degrades the intensity of the image pixel. In terms of transfer curve, it represents the offset in the x-axis. The valid value ranges from 0 to 255.

DRK: Brightness adjustment coefficient

IMG+0128h Contrast adjustment coefficient IMGPROC_CONADJ

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**CON**							
Type									R/W							
Reset									20h							

This register defines the parameter of contrast adjustment for the image pixel. The parameter is in unsigned format with normalization factor 20h. Setting the value to be greater than 20h enhances the contrast for the image; and setting the value to be less than 20h lowers the contrast for the image. The valid value ranges from 0 to 255.

 © 2009 MediaTek Inc.

CON: Contrast adjustment coefficient

IMG+0130h Colorize u component coefficient IMGPROC_COLORIZEU

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**UCOM**							
Type									R/W							
Reset									0							

IMG+0134h Colorize v component coefficient IMGPROC_COLORIZEV

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**VCOM**							
Type									R/W							
Reset									0							

These registers controls the parameters of colorize effect for the image. The valid value ranges from –128 to 127 in 2's complement format. If the values of both coefficients are zero, it implies the gray-scale effect.

UCOM: Colorize effect u component coefficient.
VCOM: Colorize effect v component coefficient.

IMG+0170h Gamma correction offset value for segment 0 IMGPROC_GAMMA_OFF0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**OFF0**							
Type									R/W							
Reset									0							

This register stores the y-offset value of the segment 0 for gamma correction.

OFF0: Offset value

For offset values of other segments, please refer to **Table 109**.

IMG+0190h Gamma correction slope value for segment 0 IMGPROC_GAMMA_SLP0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SLP0**							
Type									R/W							
Reset									0							

This register stores the slope value of the segment 0 for gamma correction.

SLP0: Slope value

For slope values of other segments, please refer to **Table 109**.

© 2009 MediaTek Inc.

IMG+01B0h Gamma correction control register IMGPROC_GAMMA_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**GTO**
Type																R/W
Reset																0

This register is used to control the gamma correction mode.

GTO: Indicator of gamma value

0 Gamma value is not greater than one.

1 Gamma value is greater than one.

Register Address	Register Function	Acronym
IMG+0170h	Offset value for the 1st segment	IMGPROC_GAMMA_OFF0
IMG+0174h	Offset value for the 2nd segment	IMGPROC_GAMMA_OFF1
IMG+0178h	Offset value for the 3rd segment	IMGPROC_GAMMA_OFF2
IMG+017Ch	Offset value for the 4th segment	IMGPROC_GAMMA_OFF3
IMG+0180h	Offset value for the 5th segment	IMGPROC_GAMMA_OFF4
IMG+0184h	Offset value for the 6th segment	IMGPROC_GAMMA_OFF5
IMG+0188h	Offset value for the 7th segment	IMGPROC_GAMMA_OFF6
IMG+018Ch	Offset value for the 8th segment	IMGPROC_GAMMA_OFF7
IMG+0190h	Slope value for the 1st segment	IMGPROC_GAMMA_SLP0
IMG+0194h	Slope value for the 2nd segment	IMGPROC_GAMMA_SLP1
IMG+0198h	Slope value for the 3rd segment	IMGPROC_GAMMA_SLP2
IMG+019Ch	Slope value for the 4th segment	IMGPROC_GAMMA_SLP3
IMG+01A0h	Slope value for the 5th segment	IMGPROC_GAMMA_SLP4
IMG+01A4h	Slope value for the 6th segment	IMGPROC_GAMMA_SLP5
IMG+01A8h	Slope value for the 7th segment	IMGPROC_GAMMA_SLP6
IMG+01ACh	Slope value for the 8th segment	IMGPROC_GAMMA_SLP7

Table 109 ***Gamma correction offset and slope register list***

IMG+0200h Color adjustment offset x for 2nd segment, red IMGPROC_COLOR1R_OFFX

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**OFFX**						
Type										R/W						
Reset										0						

IMG+0220h Color adjustment offset y for 2nd segment, red — IMGPROC_COLOR1R_OFFY

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										OFFY						
Type										R/W						
Reset										0						

IMG+0240h Color adjustment slope for 2nd segment, red — IMGPROC_COLOR1R_SLP

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											SLP					
Type											R/W					
Reset											0					

The above lists part of the registers that define the color adjustment parameters.
Color adjustment in Image Engine is used to tune the red, green, and blue color dimension individually to exhibit required color tone as a whole. The image engine provides 3-segment piecewise linear transfer curve for the user to be configured.
The x offset defines the separation point for input color value. The y offset defines the offset for each segment. The slope defines the contrast enhancement ratio for each segment.
For the red and blue components, the bit-width of the offset value is 5. For green, the bit-width of the offset value is 6. For slope of 3 color components, the bit-width is 6.

OFFX: The separation point of the input value
OFFY: The offset of output value
SLP: The slope which is used to do contrast tuning ratio within the segment.

For all the registers of color adjustment, please refer to **Table 6** for detail information.

Register Address	Register Function	Bit-width	Acronym
IMG+0200h	Color adjustment offset x for 2nd segment, red	8	IMGPROC_COLOR1R_OFFX
IMG+0204h	Color adjustment offset x for 3rd segment, red	8	IMGPROC_COLOR2R_OFFX
IMG+0208h	Color adjustment offset x for 2nd segment, green	8	IMGPROC_COLOR1G_OFFX
IMG+020Ch	Color adjustment offset x for 3rd segment, green	8	IMGPROC_COLOR2G_OFFX
IMG+0210h	Color adjustment offset x for 2nd segment, blue	8	IMGPROC_COLOR1R_OFFX
IMG+0214h	Color adjustment offset x for 3rd segment, blue	8	IMGPROC_COLOR2R_OFFX
IMG+0220h	Color adjustment offset y for 2nd segment, red	8	IMGPROC_COLOR1R_OFFY
IMG+0224h	Color adjustment offset y for 3rd	8	IMGPROC_COLOR2R_OFFY

	segment, red		
IMG+0228h	Color adjustment offset y for 2nd segment, green	8	IMGPROC_COLOR1G_OFFY
IMG+022Ch	Color adjustment offset y for 3rd segment, green	8	IMGPROC_COLOR2G_OFFY
IMG+0230h	Color adjustment offset y for 2nd segment, blue	8	IMGPROC_COLOR1R_OFFY
IMG+0234h	Color adjustment offset y for 3rd segment, blue	8	IMGPROC_COLOR2R_OFFY
IMG+0240h	Color adjustment slope for 1st segment, red	6	IMGPROC_COLOR0R_SLOPE
IMG+0244h	Color adjustment slope for 2nd segment, red	6	IMGPROC_COLOR1R_SLOPE
IMG+0248h	Color adjustment slope for 3rd segment, red	6	IMGPROC_COLOR1R_SLOPE
IMG+0250h	Color adjustment slope for 1st segment, green	6	IMGPROC_COLOR0G_SLOPE
IMG+0254h	Color adjustment slope for 2nd segment, green	6	IMGPROC_COLOR1G_SLOPE
IMG+0258h	Color adjustment slope for 3rd segment, green	6	IMGPROC_COLOR1G_SLOPE
IMG+0260h	Color adjustment slope for 1st segment, blue	6	IMGPROC_COLOR0B_SLOPE
IMG+0264h	Color adjustment slope for 2nd segment, blue	6	IMGPROC_COLOR1B_SLOPE
IMG+0268h	Color adjustment slope for 3rd segment, blue	6	IMGPROC_COLOR1B_SLOPE

Table 110 ***Color adjustment offset and slope register list***

IMG+0318h R2Y source select IMGPROC_R2Y_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												R2Y0_IO_SEL[4:0]				
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					Y2R1_IO_SEL[3:0]					IPP_IO_SEL[6:0]						
Type					R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0		0	0	0	0	0	0	0

IMG+0320h Imgproc machine enable IMGPROC_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name				RST				Y2R1_EN				R2Y0_EN				IPP_EN
Type				R/W				R/W				R/W				R/W
Reset				1				0				0				0

This register defines the enable and reset signals of the image processor.

RST: The reset bit of IPP module

- 0 Reset all status excluding control registers
- 1 Enable normal function.

Y2R1_EN: The hardware enable bit of the Y2R1 module

R2Y0_EN: The hardware enable bit of the R2Y0 module

IPP_EN: The hardware enable bit of the IPP module

Note: All of the enable bits are active high, i.e. 1 to enable.

IMG+0324h IPP RGB value detection IMGPROC_IPP_RGB_DETECT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RED[7:0]							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN[7:0]								BLUE[7:0]							
Type	R/W								R/W							

This register defines the RGB value for color detection in IPP path.

IMG+0328h IPP RGB value replacement IMGPROC_IPP_RGB_REPLACE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RED[7:0]							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN[7:0]								BLUE[7:0]							
Type	R/W								R/W							

This register defines the RGB value for replacement when the detection is hit. (IPP path)

IMG+032Ch Y2R1 RGB value detection IMGPROC_Y2R1_RGB_DETECT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RED[7:0]							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN[7:0]								BLUE[7:0]							
Type	R/W								R/W							

This register defines the RGB value for detection in Y2R1 path.

IMG+0330h Y2R1 RGB value replacement IMGPROC_Y2R1_RGB_REPLACE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RED[7:0]							
Type									R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN[7:0]								BLUE[7:0]							
Type	R/W								R/W							

This register defines the RGB value for replacement when the detection is hit. (Y2R1 path)

5.17.2 Image effect application

The Image Engine is the hardware coprocessor that performs image effects on video stream or stand-alone image. It provides the following effects:

1. Hue adjustment.
2. Saturation adjustment.
3. Contrast and intensity adjustment.
4. Grayscale and colorization.
5. Gamma correction.
6. Color adjustment.
7. Spatial dithering

The format of the coefficients is listed in **Table 111**.

Function	***Parameter group***	Range (normalized factor)	Format
Hue	***C11, C12, C21, C22***	-64 ~ 64 (64)	2's complement
Saturation	SAT	0~127 (32)	Unsigned
Contrast and brightness	BRI1	0~255	Unsigned
	BRI2	0~255	Unsigned
	Contrast	0~255 (32)	Unsigned
Colorize	U, V	-128~127	2's complement
Gamma correction	Offset	0~63	Unsigned
	Slope	0~255 (16)	Unsigned
Color adjustment	Offset for red	0~31	Unsigned
	Slope for red	0~63 (16)	Unsigned
	Offset for green	0~63	Unsigned
	Slope for green	0~63 (16)	Unsigned

	Offset for blue	0~31	Unsigned
	Slope for blue	0~63 (16)	Unsigned

Table 111 ***Coefficients format table***

5.17.2.1 Gamma correction and color adjustment

Gamma correction is a nonlinear technique. Image engine uses linear-approximation scheme for it and the same curve is applied equally on red, green, and blue components.

Two approaches are provided. For the first one, the overall input value is equally divided into 8 segments. It's suitable for the case when gamma is greater than 1. For the second one, the value is divided into 6 unsymmetrical segments. It's suitable for the case when gamma is smaller then 1.

Color adjustment is used to adjust different colors with different curves. For each color, a 3 segment piece-wise linear curve is applied. The user has to decide the offsets and the slopes of these 3 segments. The coefficients should be positive.

Cool tone and warm tone filters are both popular applications for color adjustment. Examples of gamma correction and color adjustment are as follows.

5.18 JPEG Decoder

5.18.1 General Descriptions

Now most images must be stored as JPEG format compressed files. In order to display this kind of file and boost image processing performance, the hardware JPEG decoder is developed. As a result, JPEG Decoder is designed to decode all baseline and progressive JPEG images with all YUV sampling frequencies combinations.

To gain the best speed performance, JPEG decoder will handle all portions of JPEG files except the 17-byte SOF marker. The software program only needs to program related control registers based on the SOF marker and wait for an interrupt coming from hardware. ***Fig 1*** shows the basic JPEG file structure and starting address that JPEG decoder needs. The information of DQT and DHT table is included in the JPEG file but need to be parsed by the JPEG decoder and store in the memory. The software program must program 2K-byte-align address in the table starting address because we has fixed the locations of all kinds of tables, as shown in ***Fig 2****.*

Taking into consideration the limited size of memories, hardware also supports multiple runs of JPEG progressive images and breakpoints insertion in huge JPEG files. Multiple runs can greatly reduce memory usage by 1/N where N is the number of runs. Before starting the current decoding run, JPEG decoder must reload the control signals from table address, as shown in ***Fig 2***, and then start the decoding. So the starting table address can't be changed if the multiple-run of one progressive JPEG file is enabled. Breakpoints insertion allows software to load partial JPEG file from external flash to internal memory if the JPEG file is too large to sit internally at one time.

JPEG decoder can support to only decode the target range in one baseline JPEG file. If enable this function, the decoding operations in un-targeted range will be skipped, as shown in ***Fig 3.***

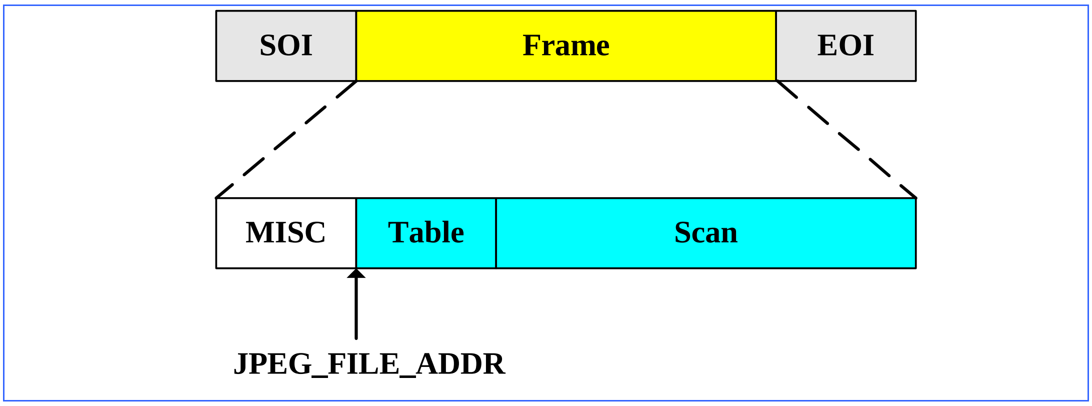


Fig 1 The basic structure of JPEG files.

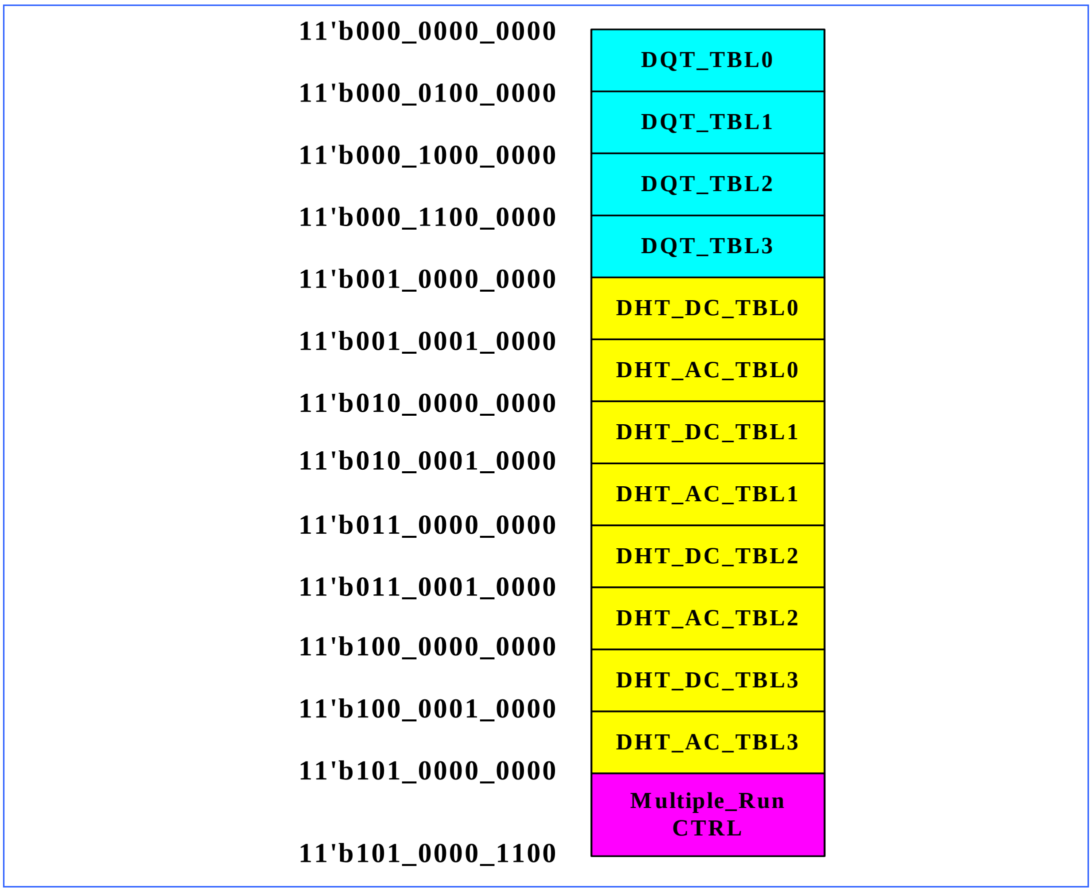


Fig 2 The memory address arrangement of DQT, DHT.

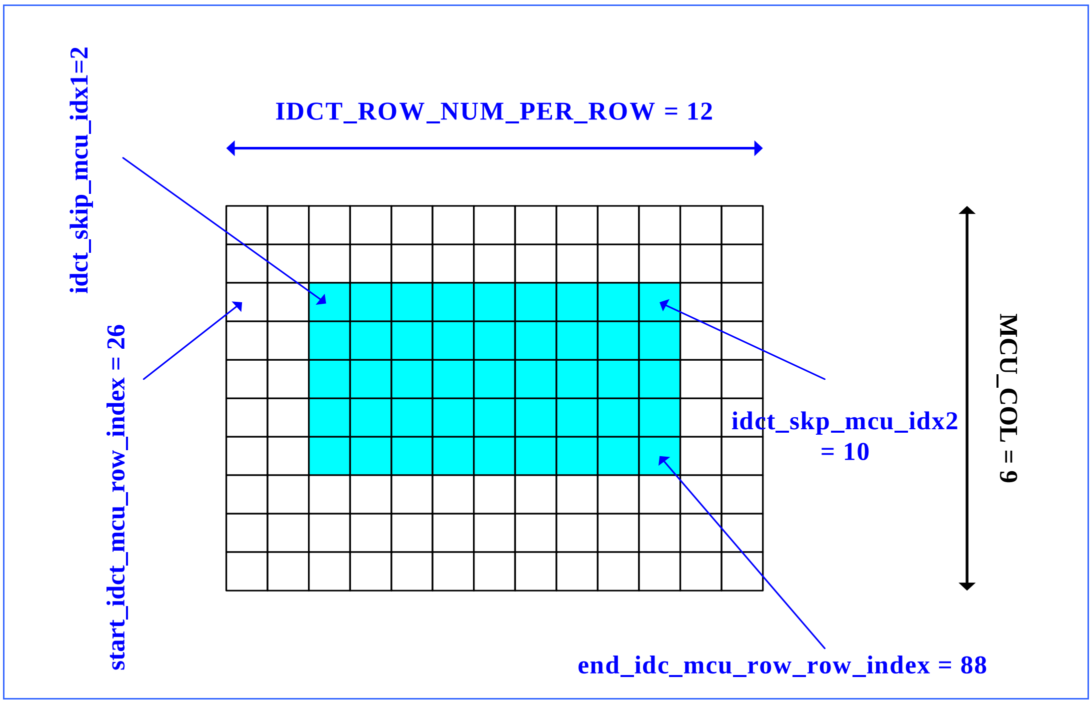


Fig 3 The range setting for the range decoding.

5.18.2 Register Definitions

Register Address	Register Function	Acronym
JPEG + 0000h	JPEG decoder file address register	**JPG_DEC_FILE_ADDR**
JPEG + 0004h	JPEG decoder table address register	**JPG_DEC_TBL_ADDR**
JPEG + 0008h	JPEG decoder sampling factor register	**JPG_DEC_SAMP_FACTOR**
JPEG + 000Ch	JPEG decoder component ID register	**JPG_DEC_COMP_ID**
JPEG + 0010h	JPEG decoder total mcu number register	**JPG_DEC_TOTAL_MCU_NUM**
JPEG + 0014h	JPEG decoder interleave mcu number per mcu row register	**JPG_DEC_MCU_NUM_PER_ MCU_ROW**
JPEG + 0018h	JPEG decoder 0st component's non-interleave data-unit number per mcu row register	**JPG_DEC_COMP0_NOINTV_ DU_NUM_PER_ MCU_ROW**
JPEG + 011Ch	JPEG decoder 1st component's non-interleave data-unit number per mcu row register	**JPG_DEC_COMP1_NOINTV_ DU_NUM_PER_ MCU_ROW**
JPEG + 0020h	JPEG decoder 2nd component's non-interleave data-unit number per mcu row register	**JPG_DEC_COMP2_NOINTV_ DU_NUM_PER_ MCU_ROW**
JPEG + 0024h	JPEG decoder 0st component's data-unit number register	**JPG_DEC_COMP0_DU_NUM**

© 2009 MediaTek Inc.

JPEG + 0028h	JPEG decoder 1st component's data-unit number register	**JPG_DEC_COMP1_DU_NUM**
JPEG + 002Ch	JPEG decoder 2nd component's data-unit number register	**JPG_DEC_COMP2_DU_NUM**
JPEG + 0030h	JPEG decoder 0st component's progressive coefficient register	**JPG_DEC_COMP0_PROG_ COEF_ADDR**
JPEG + 0034h	JPEG decoder 1st component's progressive coefficient register	**JPG_DEC_COMP1_PROG_ COEF_ADDR**
JPEG + 0038h	JPEG decoder 2nd component's progressive coefficient register	**JPG_DEC_COMP2_PROG_ COEF_ADDR**
JPEG + 003Ch	JPEG decoder control register	**JPG_DEC_CTRL**
JPEG + 0040h	JPEG decoder trigger register	**JPG_DEC_TRIG**
JPEG + 0044h	JPEG decoder reset register	**JPG_DEC_RSTB**
JPEG + 0048h	JPEG decoder breakpoint address register	**JPG_DEC_BRP_ADDR**
JPEG + 004Ch	JPEG decoder file size register	**JPG_DEC_FILE_SIZE**
JPEG + 0050h	JPEG decoder interleave first mcu index register	**JPG_DEC_INTLV_FIRST_ MCU_ IDX**
JPEG + 0054h	JPEG decoder interleave last mcu index register	**JPG_DEC_LAST_LAST_ MCU_ IDX**
JPEG + 0058h	JPEG decoder non-interleave 0st compont's first data unit index register	**JPG_DEC_NOINTV _COMP0_ FIRST_DU_ IDX**
JPEG + 005Ch	JPEG decoder non-interleave 0st compont's last data unit index register	**JPG_DEC_NOINTV _COMP0_ LAST_DU_ IDX**
JPEG + 0060h	JPEG decoder non-interleave 1st compont's first data unit index register	**JPG_DEC_NOINTV _COMP1_ FIRST_DU_ IDX**
JPEG + 0064h	JPEG decoder non-interleave 1st compont's last data unit index register	**JPG_DEC_NOINTV _COMP1_ LAST_DU_ IDX**
JPEG + 0068h	JPEG decoder non-interleave 2nd compont's first data unit index register	**JPG_DEC_NOINTV _COMP2_ FIRST_DU_ IDX**
JPEG + 006Ch	JPEG decoder non-interleave 2nd compont's last data unit index register	**JPG_DEC_NOINTV _COMP2_ LAST_DU_ IDX**
JPEG + 0070h	JPEG decoder Quantization table ID register	**JPG_DEC_QT_ID**
JPEG + 0074h	JPEG decoder interrupt status register	**JPG_DEC_INT_STATUS**
JPEG + 0078h	JPEG decoder FSM status register	**JPG_DEC_FSM_STATUS**
JPEG + 007Ch	JPEG decoder range decode register	**JPG_DEC_RANGE_EN**
JPEG + 0080h	JPEG decoder range decoding starting mcu row index register	**JPG_DEC_RANGE_START_ MCU_IDX**
JPEG + 0084h	JPEG decoder range decoding ending mcu row index register	**JPG_DEC_RANGE_END_ MCU_IDX**
JPEG + 0088h	JPEG decoder range decoding mcu number pre row register	**JPG_DEC_RANGE_MCU_ NUM_PER_ROW**

JPEG + 008Ch	JPEG decoder range decoding skipping mcu index1 register	**JPG_DEC_RANGE_SKIP_ MCU_ IDX1**
JPEG + 0090h	JPEG decoder range decoding skipping mcu index2 register	**JPG_DEC_RANGE_SKIP_ MCU_ IDX2**
JPEG + 0094h	JPEG decoder mcu counts register	**JPG_DEC_MCU_CNT**
JPEG + 0098h	JPEG decoder IDCT counts register	**JPG_DEC_IDCT_CNT**

Table 112 JPEG decoder Registers

5.18.3 Register Definitions

JPEG+0000h JPEG decoder File address register JPG_DEC_FILE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FILE_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FILE_ADDR[15:2]														0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO

FILE_ADDR Starting address of input JPEG file.

The JPEG file starting address must be a 4-byte aligned. This register’s content will be changed by hardware if JPEG decoder has been started. The contents represent the address that has been read. Suggest the starting address should be set to the location of DQT marker (0xffdb) in the beginning of decoding one JPEG file. Some dummy data can be inserted before the JPEG file and this can help to meet the limitation of 4-bytes align. When enabling the function of breakpoint, one JPEG file can be divided into several sections and then load to memory for decoding individually. In the every ending of section, software need to re-programmed the file address to the starting the next file section. It’s not permitted that any dummy data is inserted in file content. Not affected by global reset and JPEG decoder abort.

JPEG+0004h JPEG decoder table address register JPG_DEC_TBL_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TBL_ADDR[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TBL_ADDR[15:11]															
Type	R/W	R/W	R/W	R/W	R/W											

TBL_ADDR The starting address of the memory space for 4 quantization tables and 8 Huffman tables. The memory address must be a **multiple of 2K Bytes**.

The table starting address must be a multiple of 2K. The JPEG decoder will parser the information of DQT and DHT table and store in this memory location, as shown in ***Fig 2.*** Not affected by global reset and JPEG decoder abort. Need reprogramming and keep the same address in all runs of progressive images.

JPEG+0008h JPEG decoder sample factor register JPG_DEC_SAMP_FACTOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					H_SAMP_0[1:0]		V_SAMP_0[1:0]		H_SAMP_1[1:0]		V_SAMP_1[1:0]		H_SAMP_2[1:0]		V_SAMP_2[1:0]	
Type					R/W		R/W		R/W		R/W		R/W		R/W	

H_SAMP_0 Horizontal sampling factor of the 1st component, Y.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- 11 SF is 4

V_SAMP_0 Vertical sampling factor of the 1st component, Y.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- **11** SF is 4

H_SAMP_1 Horizontal sampling factor of the 2nd component, U.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- **11** SF is 4

V_SAMP_1 Vertical sampling factor of the 2nd component, U.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- 11 SF is 4

H_SAMP_2 Horizontal sampling factor of the 3rd component, V.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- **11** SF is 4

V_SAMP_2 Vertical sampling factor of the 3rd component, V.
- **00** SF is 1
- **01** SF is 2
- **10** Invalid
- **11** SF is 4

This register contains the sampling factor of YUV components. For the grayscale JPEG file, the sampling factor only has one kind of setting, ***(H0, V0)= (1, 1)***. Not affected by global reset and JPEG decoder abort.

© 2009 MediaTek Inc.

JPEG+000Ch JPEG decoder component ID register JPG_DEC_COMP_ID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMP0_ID[7:0]								COMP1_ID[7:0]							
Type	R/W								R/W							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COMP2_ID[7:0]															
Type	R/W															

COMP0_ID The 1st component (Y) ID extracted from SOF marker.
COMP1_ID The 2nd component (U) ID extracted from SOF marker.
COMP2_ID The 3rd component (V) ID extracted from SOF marker.
This register contains the IDs of YUV components. Not affected by global reset and JPEG decoder abort.

JPEG+0010h JPEG decoder total mcu number register JPG_DEC_TOTAL_MCU_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TOTAL_MCU_NUM[31:16]															
Type	R/W															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TOTAL_MCU_NUM[15:0]															
Type	R/W															

TOTAL_MCU_NUM Total MCU number will be decoded.
This register contains the total MCU number in interleaved scan. Note that if the MCU number is ***N***, program ***(N-1)*** into this register. Not affected by global reset and JPEG decoder abort.

JPEG+0014h JPEG decoder interleave mcu number per mcu row register JPG_DEC_INTLV_MCU_NUM_PER_MCU_ROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							INTLV_MCU_NUM_PER_MCU_ROW[9:0]									
Type							R/W									

INTLV_MCU_NUM_PER_ROW MCU number in the interleave scan of progressive JPEG files.
Only effective in progressive images. Note that if the MCU number per row in interleaved scan is ***N***, program ***N*** into this register. ***N*** affected by global reset and JPEG decoder abort.

JPEG+0018h JPEG decoder non-interleave 0st compont's data number setting register JPG_DEC_COMP0_NONINTLV_DU_NUM_PER_MCU_ROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DUMMY_DU				COMP0_NONINTLV_MCU_NUM_PER_MCU_ROW[9:0]									
Type			R/W				R/W									

DUMMY_DU Dummy data unit number in non-interleaved scan of the 0st component

- **00** no dummy data unit
- **01** one dummy data unit
- **10** two dummy data units
- **11** three dummy data units

COMP0_NONINTLV_MCU_NUM_PER_MCU_ROW The MCU number per row in non-interleaved scan of the 1st component (Y).

Only effective in progressive images. This register contains the MCU number per row in non-interleaved scan of the 0st component (Y). Not affected by global reset and JPEG decoder abort.

In progressive image, dummy data unit columns are inevitable if more than 8 redundant pixel columns are transmitted to fill up the last MCU in a MCU row. For example, in 420 formats, a MCU is composed of 16 x 16 pixels. If a given image size is 355 x 400, for JPEG encoder to compress, the image will grow to 368 x 400 first such that both width and height are multiples of 16. It can be seen that to be divisible by 16, there are 13 redundant Y-component pixels in the horizontal (width) direction. These 13 Y-component pixels will be compressed by encoders in interleaved scans because a complete MCU will need 16 x 16 pixels. It is different from non-interleaved scans, because in non-interleaved scans a complete MCU only needs 8 x 8 Y-component pixels. Therefore, among the 13 redundant pixels the first 5 will still be compressed as interleaved scans while the last 8 will be dropped. In this case, software must program the DUMMY_DU field to 1 so the hardware will know one 8 x 8 data unit should be skipped at the last of a MCU row in non-interleaved scan.

JPEG+001Ch JPEG decoder non-interleave 1st compont's data number setting register JPG_DEC_COMP1_NONINTLV_DU_NUM_PER_MCU_ROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DUMMY_DU				COMP1_NONINTLV_MCU_NUM_PER_MCU_ROW[9:0]									
Type			R/W				R/W									

DUMMY_DU Dummy data unit number in non-interleaved scan of the 2nd component

- **00** no dummy data unit
- **01** one dummy data unit
- **10** two dummy data units
- **11** three dummy data units

COMP1_NONINTLV_MCU_NUM_PER_MCU_ROW The MCU number per row in non-interleaved scan of the 1st component (U).

Only effective in progressive images. This register contains the MCU number per row in non-interleaved scan of the 1st component (Y).Not affected by global reset and JPEG decoder abort.

JPEG+0020h JPEG decoder non-interleave compont2's data number setting register JPG_DEC_COMP2_NONINTLV_DU_NUM_PER_MCU_ROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DUMMY_DU				COMP2_NONINTLV_MCU_NUM_PER_MCU_ROW[9:0]									
Type			R/W				R/W									

DUMMY_DU Dummy data unit number in non-interleaved scan of the 2nd component

- **00** no dummy data unit
- **01** one dummy data unit
- **10** two dummy data units
- **11** three dummy data units

COMP2_NONINTLV_MCU_NUM_PER_MCU_ROW The MCU number per row in non-interleaved scan of the 3rd component (V).

Only effective in progressive images. This register contains the MCU number per row in non-interleaved scan of the 2nd component (V). Not affected by global reset and JPEG decoder abort.

JPEG+0024h JPEG decoder compont0's data number register JPG_DEC_COMP0_DU_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMP0_DATA_UNIT_NUM[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP0_DATA_UNIT_NUM[15:0]															
Type	R/W															

COMP0_DATA_UNIT_NUM Data unit number of 0st component in progressive JPEG file.

Number of 0st component is:

$$(MCU_PER_ROW * H_0 - dumm_ydu_row) * (MCU_PER_Col * V_0 - dumm_ydu_col) - 1$$

Only effective in progressive images or grayscale mode. This register contains the 8x8 data unit number of the 1st component in non-interleaved scans. Note that if the data unit number is ***N***, program ***(N-1)*** into this register. Not affected by global reset and JPEG decoder abort.

JPEG+0028h JPEG decoder compont1's data number register JPG_DEC_COMP1_DU_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMP1_DATA_UNIT_NUM[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP1_DATA_UNIT_NUM[15:0]															
Type	R/W															

COMP1_DATA_UNIT_NUM Data unit number of 1st component in progressive JPEG file.

© 2009 MediaTek Inc.

Number of 1st component is:

$$TOTAL_MCU_NUM - 1$$

Only effective in progressive images. This register contains the 8x8 data unit number of the 1st component in non-interleaved frame. Note that if the data unit number is ***N***, program ***(N-1)*** into this register. Not affected by global reset and JPEG decoder abort.

JPEG+002Ch JPEG decoder compont2's data number register JPG_DEC_COMP2_DU_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMP2_DATA_UNIT_NUM[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP2_DATA_UNIT_NUM[15:0]															
Type	R/W															

COMP2_DATA_UNIT_NUM Data unit number of 2nd component in progressive JPEG file.

Number of 2nd component is:

$$TOTAL_MCU_NUM - 1$$

Only effective in progressive images. This register contains the 8x8 data unit number of the 2nd component in non-interleaved frame. Note that if the data unit number is ***N***, program ***(N-1)*** into this register. Not affected by global reset and JPEG decoder abort.

JPEG+0030h JPEG decoder compont0's progressive coefficient address register JPG_DEC_COMP0_PROG_COEFF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMP0_PROGR_COEFF_ ADDR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP0_PROGR_COEFF_ ADDR[15:0]															
Type	R/W															

COMP0_PROGR_COEFF_ADDR Destination address of 0st component's coefficient in progressive JPEG file

Only effective in progressive images. This register contains the starting address of the memory space storing the intermediate progressive 16-bit coefficients of the 1st component. The memory requirement of 1st component is

$$64 \times total_block_in_this_runx2,$$

total block means the block number of 0st component

This value must be a multiple of 4. Not affected by global reset and JPEG decoder abort.

JPEG+0034h JPEG decoder compont1's progressive coefficient address register JPG_DEC_COMP1_PROG_COEFF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	**COMP1_PROGR_COEFF_ ADDR[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COMP1_PROGR_COEFF_ ADDR[15:0]**															
Type	R/W															

COMP1_PROGR_COEFF_ADDR Destination address of 1st component's coefficient in progressive JPEG file

Only effective in progressive images. This register contains the starting address of the memory space storing the intermediate progressive 16-bit coefficients of the 2nd component. The memory requirement of 1st component is

$$64 \times total_block_in_this_runx2,$$

total block means the block number of 2nd component

This value must be a multiple of 4. Not affected by global reset and JPEG decoder abort.

JPEG+0038h JPEG decoder compont2's progressive coefficient address register JPG_DEC_COMP2_PROG_COEFF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COMP2_PROGR_COEFF_ ADDR[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COMP2_PROGR_COEFF_ ADDR[15:0]**															
Type	R/W															

COMP2_PROGR_COEFF_ADDR Destination address of 2nd component's coefficient in progressive JPEG file

Only effective in progressive images. This register contains the starting address of the memory space storing the intermediate progressive 16-bit coefficients of the 2nd component. The memory requirement of 3rd component is

$$64 \times total_block_in_this_runx2,$$

total block means the block number of 3rd component

This value must be a multiple of 4. Not affected by global reset and JPEG decoder abort.

JPEG+003Ch JPEG decoder control register JPG_DEC_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GRAY**	**JPEG_ MODE**	**DU9[2:0]**			**DU8[2:0]**			**DU7[2:0]**			**DU6[2:0]**			**DU5[2:0]**	
Type	R/W	R/W	R/W			R/W			R/W			R/W			R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**DU4[2:0]**			**DU3[2:0]**			**DU2[2:0]**			**DU1[2:0]**			**DU0[2:0]**		
Type		R/W			R/W			R/W			R/W			R/W		

This register contains 2 kinds of information: the operating mode of JPEG decoder and the order of 3 components in a MCU. Affected by global reset and JPEG decoder abort. Need reprogramming for multiple runs of progressive images.

GRAY_MODE Set the grayscale mode.

0 No-Grayscale mode
1 Grayscale mode

JPEG_MODE The operating mode of JPEG decoder.

0 Baseline mode
1 Progressive mode

DU9 The 10th data unit component category in a MCU

100 The 10th data unit is the 1st component (Y)
101 The 10th data unit is the 2nd component (U)
110 The 10th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU8 The 9th data unit component category in a MCU

100 The 9th data unit is the 1st component (Y)
101 The 9th data unit is the 2nd component (U)
110 The 9th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU7 The 8th data unit component category in a MCU

100 The 8th data unit is the 1st component (Y)
101 The 8th data unit is the 2nd component (U)
110 The 8th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU6 The 7th data unit component category in a MCU

100 The 7th data unit is the 1st component (Y)
101 The 7th data unit is the 2nd component (U)
110 The 7th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU5 The 6th data unit component category in a MCU

100 The 6th data unit is the 1st component (Y)
101 The 6th data unit is the 2nd component (U)
110 The 6th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU4 The 5th data unit component category in a MCU

100 The 5th data unit is the 1st component (Y)
101 The 5th data unit is the 2nd component (U)
110 The 5th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU3 The 4th data unit component category in a MCU

100 The 4th data unit is the 1st component (Y)

101 The 4th data unit is the 2nd component (U)
110 The 4th data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU2 The 3rd data unit component category in a MCU
100 The 3rd data unit is the 1st component (Y)
101 The 3rd data unit is the 2nd component (U)
110 The 3rd data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU1 The 2nd data unit component category in a MCU
100 The 2nd data unit is the 1st component (Y)
101 The 2nd data unit is the 2nd component (U)
110 The 2nd data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

DU0 The 1st data unit component category in a MCU
100 The 1st data unit is the 1st component (Y)
101 The 1st data unit is the 2nd component (U)
110 The 1st data unit is the 3rd component (V)
111 Not used in current frame
000-011 Invalid

JPEG+0040h JPEG decoder trigger register JPG_DEC_TRIG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**TRIG**
Type																W/O

TRIG Set to 1 to start JPEG decoder.

JPEG+0044h JPEG decoder reset register JPG_DEC_RSTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RSTB**
Type																R/W

RSTB Low active reset. Set to 0 to reset the JPEG decoder and IDCT processor. And set to 1 to finish the reset process.

JPEG Decoder reset must be issued once after finishing JPEG decoding. If not, the shared designs with MP4 can't work normally.

JPEG+0048h JPEG decoder breakpoint address register — JPG_DEC_BRP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JPEG_FILE_BRP_ADDR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JPEG_FILE_BRP_ADDR [15:2]														0	0
Type	R/W														RO	RO

JPEG_FILE_BRP_ADDR Breakpoint address. This address should be the ending of every section of JPEG files.

JPEG_DEC_BRP stands for a 32-bit byte breakpoint address that hardware will stall once the breakpoint address is encountered. The Data in the current breakpoint address will not be read. This control register provides a solution for software to swap internal memory content with external memory in case the JPEG source file is too big for internal memory to store at one time. A breakpoint interrupt will fire when hardware DMA address hits the breakpoint address. Software can refill the residue bitstream and written again these two registers, **JPG_DEC_BRP_ADDR** and **JPEG_DEC_FILE_ADDR** to re-start JPEG decoding. **JPG_DEC_BRP_ADDR must be written before JPEG_DEC_FILE_ADDR. JPEG Decoder will be re-started as JPEG_DEC_FILE_ADDR is written.** Note that the breakpoint address must be a multiple of 4 and 16-byte dummy size must be included in this run. Not affected by global reset and JPEG decoder abort.

JPEG+004Ch JPEG decoder file size register — JPG_DEC_FILE_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JPEG_FILE_TOTAL_SIZE[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JPEG_FILE_TOTAL_SIZE[15:0]															
Type	R/W															

JPEG_FILE_TOTAL_SIZE JPEG file bytes size. If JPEG file size =M, this register is should be programmed (M+64).

JPEG_FILE_TOTAL_SIZE represents the JPEG source file size in bytes. Hardware will fire a file overflow interrupt and stall if the DMA address equals to this address. Note that the file total size must be a multiple of 4 and must include 16-byte dummy size. If the file total size is not divisible by 4, increment the size value until it is. Not affected by global reset and JPEG decoder abort.

JPEG+0050h JPEG decoder interleave first mcu index register — JPG_DEC_INTLV_FIRST_MCU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													INTLV_FIRST_MCU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTLV_FIRST_MCU_INDEX[15:0]															
Type	R/W															

INTLV_FIRST_MCU_INDEX First MCU index of multiple-run decoding in interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the first MCU index that hardware will process in the interleaved scans of the current image. The JPEG decoder is able to skip certain MCUs by defining the first and last MCU index. Not affected by global reset and JPEG decoder abort.

JPEG+0054h JPEG decoder interleave last mcu index register JPG_DEC_INTLV_LAST_MCU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													INTLV_LAST_MCU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTLV_LAST_MCU_INDEX[15:0]															
Type	R/W															

INTLV_LAST_MCU_INDEX Last MCU index of multiple-run decoding in interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the last MCU index that hardware will process in the interleaved scans of the current image. The JPEG decoder is able to skip certain MCUs by defining the first and last MCU index. Not affected by global reset and JPEG decoder abort.

JPEG+0058h JPEG decoder non-interleave 0st component first data unit index register JPG_DEC_COMP0_FIRST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP0_FIRST_DU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP0_FIRST_DU_INDEX[15:0]															
Type	R/W															

COMP0_FIRST_DU_INDEX First data-unit index of multiple-run decoding 0st component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the first data unit index that hardware will process in the non-interleaved scans containing Y component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

JPEG+005Ch JPEG decoder non-interleave 0st component last data unit index register JPG_DEC_COMP0_LAST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP0_LAST_DU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP0_LAST_DU_INDEX[15:0]															
Type	R/W															

COMP0_LAST_DU_INDEX Last data-unit index of multiple-run decoding 0st component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the last data unit index that hardware will process in the non-interleaved scans containing Y component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

JPEG+0060h JPEG decoder non-interleave 1st component first data unit index register JPG_DEC_COMP1_FIRST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP1_FIRST_DU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP1_FIRST_DU_INDEX[15:0]															
Type	R/W															

COMP1_FIRST_DU_INDEX First data-unit index of multiple-run decoding 1st component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the first data unit index that hardware will process in the non-interleaved scans containing U component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

JPEG+0064h JPEG decoder non-interleave 1st component last data unit index register JPG_DEC_COMP1_LAST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP1_LAST_DU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP1_LAST_DU_INDEX[15:0]															
Type	R/W															

COMP1_LAST_DU_INDEX Last data-unit index of multiple-run decoding 1st component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the last data unit index that hardware will process in the non-interleaved scans containing U component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

JPEG+0068h JPEG decoder non-interleave 2nd component first data unit index register JPG_DEC_COMP2_FIRST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP2_FIRST_MCU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	COMP2_FIRST_DU_INDEX[15:0]
Type	R/W

COMP2_FIRST_DU_INDEX First data-unit index of multiple-run decoding 2nd component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the first data unit index that hardware will process in the non-interleaved scans containing V component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

JPEG+006Ch JPEG decoder non-interleave 2nd component last data unit index register JPG_DEC_COMP2_LAST_DU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													COMP2_LAST_DU_INDEX[19:16]			
Type													R/W			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COMP2_LAST_DU_INDEX[15:0]															
Type	R/W															

COMP2_LAST_DU_INDEX Last data-unit index of multiple-run decoding 2nd component in non-interleave scan of the progressive JPEG file.

Only effective in progressive images. This control register specifies the last data unit index that hardware will process in the non-interleaved scans containing V component of the current image. The JPEG decoder is able to skip certain data units by defining the first and last data unit index. Not affected by global reset and JPEG decoder abort.

In progressive image, interleave and no-interleave scans can both exit in one image. If we want to enable the function of multiple run, the range of every run for interleave and no-interleave scans need to be programmed simultaneously. For example, in 420 format, a MCU is composed of 16 x 16 pixels. If a image is 162x128,

If we set run range for interleave scan,

(INTLV_FIRST_MCU_INDEX, INTLV_LAST_MCU_INDEX)= (0, 20),

the run range of non-interleave scan should be

(COMP0_FIRST_DU_INDEX, COMP0_LAST_DU_INDEX)= (0, 77) ; (COMP1_FIRST_DU_INDEX, COMP1_LAST_DU_INDEX)= (0, 20) ; (COMP2_FIRST_DU_INDEX, COMP2_LAST_DU_INDEX)= (0, 20)

, and the dummy data unit should be ignored in the non-interleave scan.

JPEG+0070h JPEG decoder quantization table ID register JPG_DEC_QT_ID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					COMP0_QT_ID[3:0]				COMP1_QT_ID[3:0]				COMP2_QT_ID[3:0]			
Type					R/W				R/W				R/W			

COMP0_QT_ID Quantization table ID of Y component directly extracted from SOF marker

COMP1_QT_ID Quantization table ID of U component directly extracted from SOF marker

COMP2_QT_ID Quantization table ID of V component directly extracted from SOF marker

This register contains the quantization table IDs for YUV components. Not affected by global reset and JPEG decoder abort.

JPEG+0074h JPEG decoder interrupt status register JPEG_DEC_INT_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IRQ_TYP**															
Type	1															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**INT3**	**INT2**	**INT1**	**INT0**
Type													R/WC	R/WC	R/WC	R/WC

The register reflects the interrupt status

INT3 Set to 1 by range decoding interrupt, and write 1 to clear interrupt signal.

INT2 Set to 1 by file overflow interrupt, and write 1 to clear interrupt signal.

INT1 Set to 1 by breakpoint interrupt, and write 1 to clear interrupt signal.

INT0 Set to 1 by end of file interrupt, and write 1 to clear interrupt signal.

IRQ_TYP IRQ Type

- **0** Read clear mode
- **1** Write clear mode **(if enable write clear mode, this bit must be always set to 1)**

JPEG+0078h JPEG decoder FSM status register JPG_DEC_FSM_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**FOS**	**BRPS**	**EOFS**		**JPEG_DEC_STATE**			**HUFF_DEC_STATE**				**MARKER_PARSER_STATE**			
Type		RO	RO	RO		RO			RO				RO			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SOS_PARSER_STATE**					**DHT_PARSER_STATE**				**DQT_PARSER_STATE**			**DATA_UNIT_STATE**			
Type	RO					RO				RO			RO			

The register reflects the *FSM* status

FOS Set to 1 in overflow condition

BRPS Set to 1 in breakpoint condition

EOFS Set to 1 in EOI condition

JPEG_DEC_STATE

- **0 IDLE**
- **1 DMA_LOAD**
- **2 MARKER_PARSER**
- **3 HUFFMAN_DECODE**
- **4 RST_SRCH**
- **5 RELOAD_PROGR_COEFF**
- **6 WAIT_FOR_IDCT**

HUFF_DEC_STATE

- **0 IDLE**
- **1 RELOAD_SCAN_START_ADDR**

2 RELOAD_EOB_RUN
3 BYPASS_DATA
4 RELOAD_CTRL
5 READ_COEFF
6 HUFF_ADDR_LOGICAL
7 HUFF_ADDR_PHYSICAL
8 EOB_RUN_GEN
9 AMP_CAL
A WAIT_COEFF
B SAVE_SCAN_START_ADDR
C SAVE_EOB_RUN
D SAVE_CRL
E WAIT_FOR_IDCT
MARKER_PARSER_STATE
1 SRCH_0xFF
2 MARKER_TYPE_IDEN
3 MARKER_LEN_HIGH
4 MARKER_LEN_LOW
5 MARKER_MISC_INFO_DEC
SOS_PARSER_STATE
1 NS
2 COMP_SPEC
3 SS_FIELD
4 SE_FIELD
5 AH_AL_FIELD
DHT_PARSER_STATE
1 TC_TH
2 HUFF_SYM_LENGTH
3 HUFF_VAL_COLLECT
4 WRITE_HUFF_VAL
DQT_PARSER_STATE
0 IDLE
1 PQ_TQ
2 READ_COEFF
3 PROGR_COEFF
4 WRITE_COEFF
HUFF_DEC_STATE
0 DU0
1 DU1
2 DU2
3 DU3
4 DU4
5 DU5

6	DU6
7	DU7
8	DU8
9	DU9

JPEG+007Ch JPEG decoder range decode enable register JPG_DEC_RANGE_DEC_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																EN
Type																R/W

EN Set to 1 is to enable partial decoding.

The register is to enable the function of partial decoding. Only support the baseline JPEG file.

JPEG+0080h JPEG decoder range decode starting mcu index register JPG_DEC_RANGE_START_MCU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	START_IDCT_MCU_ROW_INDEX[15:0]															
Type	R/W															

START_IDC_MCU_ROW_INDEX MCU index of the starting row in range decoding. Please see ***Fig 3.***

The register is to set which mcu is the starting point. Note that if the MCU number is ***N***, The setting range is from 0 ~ ***(N-1)***.

JPEG+0084h JPEG decoder range decode ending mcu index register JPG_DEC_RANGE_END_MCU_IDX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	END_IDCT_MCU_ROW_INDEX															
Type	R/W															

END_IDC_MCU_ROW_INDEX MCU index of the ending row in range decoding. Please see ***Fig 3.***

The register is to set which mcu index is the ending point. Note that if the MCU number is ***N***, The setting range is from 1 ~ ***(N)***.

JPEG+0088h JPEG decoder range decode mcu number pre row register JPG_DEC_RANGE_MCU_NUM_PER_ROW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IDCT_MCU_NUM_PER_ROW															
Type	R/W															

IDC_MCU_NUM_PER_ROW MCU number per image's row. Please see ***Fig 3.***
The register is to define the valid number per row in the image. Note that if the MCU number per row is ***M***, The setting range is from 0 ~ ***(M-1)***.

JPEG+008Ch JPEG decoder range decode skip mcu index1 register JPG_DEC_RANGE_SKIP_MCU_IDX1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IDCT_SKIP_MCU_INDEX1															
Type	R/W															

IDC_SKIP_MCU_INDEX1 Skip MCU index1 pre MCU row for range decoding. Please see ***Fig 3.***
The register is to define the starting muc index per row in the image. Note that if the MCU number per row is ***M***, The setting range is from 0 ~ (***M-2***).

JPEG+0090h JPEG decoder range decode skip mcu index2 register JPG_DEC_RANGE_SKIP_MCU_IDX2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IDCT_SKIP_MCU_INDEX2															
Type	R/W															

IDC_SKIP_MCU_INDEX2 Skip MCU index2 pre MCU row for range decoding. Please see ***Fig 3.***
The register is to define the ending muc index per row in the image. Note that if the MCU number per row is ***M***, The setting range is from 1 ~ ***(M-1)***.

JPEG+0094h JPEG decoder mcu counts register JPG_DEC_MCU_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												JPG_MCU_CNT[19:16]				
Type												RO				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JPG_MCU_CNT[15:0]															
Type	RO															

JPG_MCU_CNT MCU counts that had been decoded.
The register is to report the current mcu number that has been decoded by JPEG decoder.

JPEG+0098h JPEG decoder IDCT counts register JPG_DEC_IDCT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IDCT_BLK_CNT[15:0]															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JPEG_BLK_CNT[15:0]															
Type	RO															

The register is to report the current block number that has been decoded by JPEG decoder and IDCT.

JPEG_BLK_CNT Block number that has been finished by JPEG decoder design.

IDCT_BLK_CNT Block number that has been finished by IDCT design.

5.19 JPEG Encoder

5.19.1 General Descriptions

The hardware JPEG encoder implements the baseline mode of Standard ISO/IEC 10918-1. It supports YUV 422, 420 and 411 formats for color pictures and grayscale format. With the software assist and suitable destination memory address setting; JFIF/EXIF JPEG format can also be supported. For hardware reduction, it uses standard DC and AC Huffman tables for both the luminance and chrominance components. To adjust the picture compression ratio and picture quality, there are 14 levels of quantization that can be programmed. After initialization by software, the hardware JPEG encoder can generate the entire compressed file.

Fig. 16 shows the procedure of the JPEG encoder. The YUV pixel data that came from image DMA are grouped into 8x8 blocks and then down-sampled to YUV 422, YUV420 and YUV411 format. For grayscale encoding, only Y component is present. When encoding, the first thing to do is to turn the pixel data into the frequency domain using FDCT. After the quantizer is done, the quantized DCT coefficients are encoded by RLE and VLC.

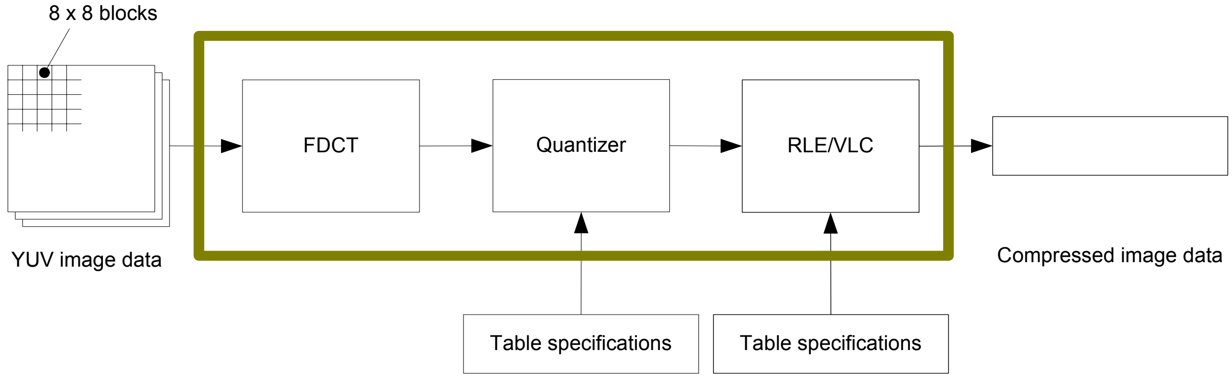


Fig. 16 The procedure of JPEG encoder.

5.19.2 JFIF/EXIF JPEG Format

The original JPEG file can contain the information of camera, the thumbnail image and some good and important information by following JFIF/EXIF format. If enabling JFIF/EXIF mode, the JPEG encoder does not generate SOI marker and related thumbnail header and small image data. The JPEG encoder just outputs

the bitstreams from DQT marker. The software programs need to provide the suitable destination address after estimating the size of SOI marker, related thumbnail header and small image data. The SOI marker and related thumbnail header need to be handled by software programs and the small image data can be output by IMGDMA. With the suitable destination address configuration, the JFIF/EXIF JPEG format can be generated. **Fig 17** illustrates the data partition for JFIF/EXIF support. Before JPEG encoding, three suitable address configurations provides. The ADDR1 is provided to SOI maker and the thumbnail header. This part is handled by software. The ADDR2 is provided to IMGDMA to write RGB small image data. The last address, ADDR3, provides to the JPEG encoder to write out remaining bitstreams.

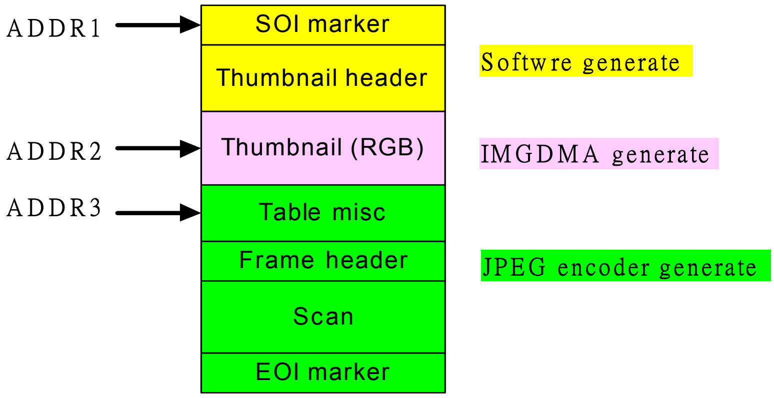


Fig 17 The JFIF/EXIF data structure.

5.19.3 Register Definitions

Register Address	Register Function	Acronym
JPEG + 0100h	JPEG encoder reset register	**JPG_ENC_RST**
JPEG + 0104h	JPEG encoder control register	**JPG_ENC_CTL**
JPEG + 0108h	JPEG encoder quality register	**JPG_ENC_QUALITY**
JPEG + 010Ch	JPEG encoder block number register	**JPG_ENC_BLK_NUM**
JPEG + 0110h	JPEG encoder block count register	**JPG_ENC_BLK_CNT**
JPEG + 0114h	JPEG encoder frame number register	**JPG_ENC_FRAME_NUM**
JPEG + 0118h	JPEG encoder frame count register	**JPG_ENC_FRAME_CNT**
JPEG + 011Ch	JPEG encoder interrupt status register	**JPG_ENC_INTSTS**
JPEG + 0120h	JPEG encoder 1st base address register	**JPG_ENC_DEST_ADDR1**
JPEG + 0124h	JPEG encoder 1st DMA address register	**JPG_ENC_DMA_ADDR1**
JPEG + 0128h	JPEG encoder 1st STALL address register	**JPG_ENC_STALL_ADDR1**
JPEG + 012Ch	JPEG encoder 2nd base address2	**JPG_ENC_DEST_ADDR2**

 © 2009 MediaTek Inc.

	register	
JPEG + 0130h	JPEG encoder 2nd DMA address2 register	**JPG_ENC_DMA_ADDR2**
JPEG + 0134h	JPEG encoder 2nd STALL address2 register	**JPG_ENC_STALL_ADDR2**
JPEG + 0138h	JPEG encoder offset address register	**JPG_ENC_OFFSET_ADDR**

Table 113 JPEG encoder Registers

JPEG+0100h JPEG encoder reset register JPG_ENC_RST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RSTB**
Type																R/W
Reset																1

RSTB Set to 0 to reset the JPEG encoder and FDCT processor. And set to 1 to finish the reset process.

JPEG+0104h JPEG encoder control register JPG_ENC_CTL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**ULTRA_EN**		**ADDR_SW**	**CONT**	**JPG**	**YUV**		**IT**	**GRAY**	**EN**
Type							R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset							0		0	0	0	0		1	0	0

EN Enable the JPEG encoder. This bit is cleared by hardware after encoding is done.

JPEG will write out the bitstream of JPEG header after this enable bit is set. After finish writing out the header stream, the YUV data request to JPEG DMA will be issued. In order to avoid some abnormal conditions, we assume that JPEG DMA must be enabled before set this enabling bit.

GRAY Do grayscale encode. Remember that the image DMA should be programmed as grayscale too.

- **0** color
- **1** grayscale

IT Interrupt enabling

- **0** Disable
- 1 Enable

YUV YUV format

- 0 YUV 422
- 1 YUV 420
- 2 Reserved
- 3 YUV 411

JPG JPEG or other application format support

- 0 JPEG
- 1 JFIF/EXIF

If enabling JFIF/EXIF format, JPEG encoder won't write out SOI marker and write out DQT maker. Software program can fill the JFIF/EXIF content above the DQT marker to finish one JFIF/EXIF JPEG file.

CONT JPEG continuous shooting.

0 OFF
1 ON
JPEG continuous shooting mode can encode continuous YUV data until it reaches the target frame number.

ADDR_SW JPEG destination address switch in continuous shooting mode
0 OFF, frame destination address will be accumulated until it reaches the target frame numbe.
1 ON, frame destination address switches between JPG_ENC_DEST_ADDR and JPG_ENC_DEST_ADDR2.

ULTRA_EN JPEG encoder accessing GMC with ultra high priority enable
0 OFF, JPEG encoder accessing GMC with normal priority.
1 ON, JPEG encoder accessing GMC with high priority if buffer reaches the threshold

JPEG+0108h JPEG encoder quality register JPG_ENC_QUALITY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**QT**		**QUALITY**	
Type													R/W		R/W	
Reset													00		00	

QUALITY Quality level in quantization tables.
00 Low, only for High and Good quality quantization table.
01 Fair.
10 Good.
11 High.

QT Quantization Table Selection
00 High quality table, 2 ~ 4 time compression ratio.
01 Good quality table, 3 ~ 6 time compression ratio.
10 Fair quality table, 5 ~ 10 time compression ratio.
11 Low quality table, 7 ~ 30 time compression ratio.

Table 114 details all kinds of quality levels that JPEG encoder can provide. Larger quality factor means better compression quality.

QT	Quality	Quality Factor
00	00	60
	01	80
	10	90
	11	95
01	00	39
	01	68
	10	84
	11	92
10	00	48
	01	48
	10	74
	11	87
11	00	34
	01	34
	10	64
	11	82

Table 114 Quality v.s. Quantization Table setting

JPEG+010Ch JPEG block number register JPG_ENC_BLK_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											BLK_NUM[21:16]					
Type											R/W					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLK_NUM[15:0]															
Type	RO															
Reset	0															

BLK_NUM 8x8 block number will be encoded.

$$BLK_NUM = ceil(Width / H_Y) \times ceil(Heigh / V_Y) \times (H_Y \times V_Y + H_U \times V_U + H_V \times V_V) - 1 \quad ,$$

Note: The dummy 8x8 block must be considered in *BLK_NUM*. For example, a 162x128 YUV420 JPEG file, the block number setting should be 527.

JPEG+0110h JPEG block count register JPG_ENC_BLK_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											BLK_CNT[21:16]					
Type											RO					
Reset											0					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLK_CNT[15:0]															
Type	RO															
Reset	0															

 © 2009 MediaTek Inc.

BLK_CNT 8x8b lock count has been encoded. CNT will increase 1 after finishing *RLE/VLC*.

JPEG+0114h JPEG encoder continuous shooting frame number JPG_ENC_FRAME_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name											**FRAME_NUM**					
Type											R/W					
Reset											0					

FRAME_NUM Frame number in continuous shooting will be encoded

JPEG+0118h JPEG encoder continuous shooting current frame count JPG_ENC_CURR_FRAME_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name											**FRAME_CNT**					
Type											R/O					
Reset											0					

FRAME_CNT Frame counts had been encoded.

JPEG+011Ch JPEG encoder interrupt status register JPG_ENC_INTSTS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name															**STALL**	**DONE**
Type															R/WC	R/WC
Reset															0	0

DONE Indicates that encoding operation is done. Need to set to 0 after reading.

STALL Indicates that encoding operation is in the stall condition. Need to set to 0 after reading but the stall condition can' cleared by this bit. The stall condition will be clear if we do the following procedure. The destination address need to be re-programmed firstly and then reprogrammed the stall address to clear the stall condition. After the stall condition is cleared, the encoding process will be started automatically.

JPEG+0120h JPEG encoder 1st base address register JPG_ENC_DEST_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	**BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BS_ADDR Base address1 of encoded data.

In the single shooting mode and continuous shooting mode without automatic address, hardware will use this address setting. In continuous shooting mode with automatic address switching, frame0, 2, 4..etc will use this address setting.

JPEG+0124h JPEG encoder 1st DMA address register JPG_ENC_DMA_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMA_ADDR[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DMA_ADDR[15:0]**															
Type	RO															
Reset	0															

CURR_ADDR The current DMA address during encoding.

In the single shooting mode, this DMA address will represent the current encoded address. But this DMA address will just represent the encoded address of fram0, 2, 4..etc in continuous shooting mode.

JPEG+0128h JPEG encoder 1st STALL address register JPG_ENC_STALL_ADDR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**STALL_ADDR[31:16]**															
Type	R/W															
Reset	32'hf															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**STALL_ADDR[15:0]**															
Type	R/W															
Reset	32'hf															

STALL_ADDR This field is the upper bound of JPEG encoder's write-address.

Note that the stall address must be word-aligned. Whenever the stall address is reached, the JPEG encoder stalls and issues an interrupt to software. After, if the software programs the JPG_ENC_STALL_ADDR to another value, the JPEG encoder resumes the encoding procedure and automatically uses the JPG_ENC_DEST_ADDR as the new starting address. It means that before we change the value of JPG_ENC_STALL_ADDR, the JPG_ENC_DEST_ADDR has to be programmed to a corresponding starting address. However, if the software wants to discard the uncompleted file, it can simply reset the JPEG encoder to cancel the encode operation. Also, it is important that the value of JPG_ENC_STALL_ADDR should be larger than JPG_ENC_DEST_ADDR by at least 608 bytes to guarantee that the header of the JPEG file can be completely written into memory.

JPEG+012Ch JPEG encoder 2nd base address register JPG_ENC_DEST_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BS_ADDR Base address of 2nd encoded data.

In the single shooting mode and continuous shooting mode without automatic address, hardware **will not** use this address setting. In continuous shooting mode with automatic address switching, frame1, 3, 5..etc will use this address setting.

JPEG+0130h JPEG encoder 2nd DMA address register JPG_ENC_DMA_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMA_ADDR2[31:16]**															
Type	RO															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DMA_ADDR2[15:0]**															
Type	RO															
Reset	0															

DMA_ADDR2 The current DMA address during 2nd encoding.

In the single shooting mode, this DMA address will represent the current encoded address. But this DMA address will just represent the encoded address of fram1, 3, 5..etc in continuous shooting mode.

JPEG+0134h JPEG encoder 2nd STALL address register JPG_ENC_STALL_ADDR2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**STALL_ADDR2[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**STALL_ADDR2[15:0]**															
Type	R/W															
Reset	0															

STALL_ADDR2 This field is the upper bound of JPEG encoder's write-address2. Note that the stall address2 only works in continuous shooting and memory auto-switch mode.

JPEG+0138h JPEG encoder Offset address register — JPG_ENC_OFFSET_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OFFSET_ADDR															
Type	R/W															
Reset	0															

OFFSET_ADDR offset address from the ending of the last frame in the continuous shooting mode. This address must be 4-byte align.

OFFSET_ADDR should be set to 0 (before enabling JPEG encoder) if we don't need this function. Because the starting destination address always use the following equation to cacluate the new starting destination JPEG encoder uses.

$$NEW_DEST_ADDR = (OFFSET_ADDR + CURR_DEST_ADDR)$$

5.20 LCD Interface

5.20.1 General Description

MT6516 contains a versatile LCD controller which is optimized for multimedia applications. This controller supports many types of LCD modules and contains a rich feature set to enhance the functionality. These features are:

- Up to 640 x 480 resolution
- The internal frame buffer supports 8bpp indexed color, RGB 565, RGB 888 and ARGB 8888 format.
- Supports 8-bpp (RGB332), 12-bpp (RGB444), 16-bpp (RGB565), 18-bit (RGB666) and 24-bit (RGB888) LCD modules.
- 6 Layers Overlay with individual color depth, window size, vertical and horizontal offset, source key, alpha value and display rotation control(90°,180°, 270°, mirror and mirror then 90°, 180° and 270°)
- One Color Look-Up Table
- Three Gamma Correction Tables
- 3x3 matrix for Color Management

For parallel LCD modules, the LCD controller can reuse external memory interface or use dedicated 8/9/16/18-bit parallel interface to access them and 8080 type interface is supported. It can transfer the display data from the internal SRAM or external SRAM/Flash Memory to the off-chip LCD modules.

For serial LCD modules, this interface performs parallel to serial conversion and both 8- and 9- bit serial interface is supported. The 8-bit serial interface uses four pins – LSCE#, LSDA, LSCK and LSA0 – to enter commands and data. Meanwhile, the 9-bit serial interface uses three pins – LSCE#, LSDA and LSCK – for the same purpose. Data read is not available with the serial interface and data entered must be 8 bits.

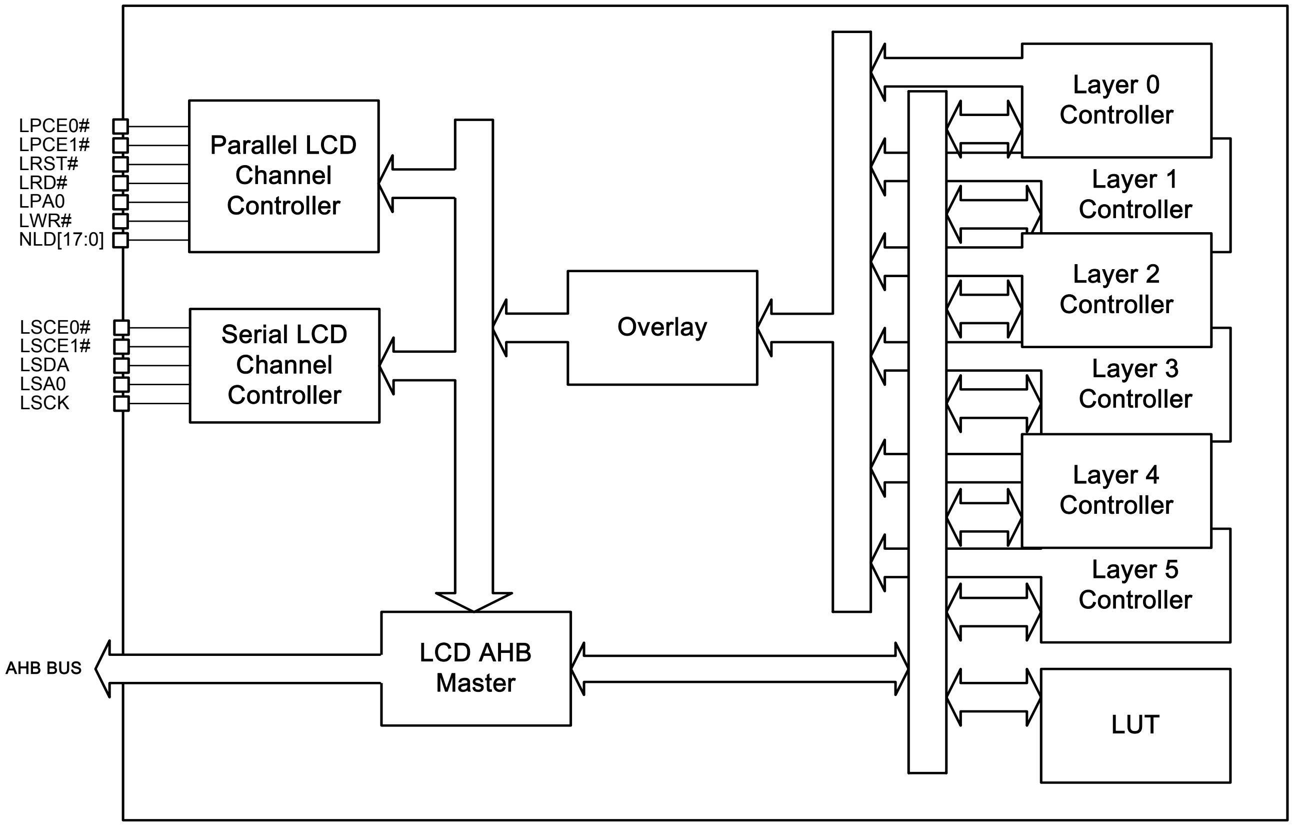


Figure 138 ***LCD Interface Block Diagram***

Figure 139 shows the timing diagram of this serial interface. When the block is idle, LSCK is forced LOW and LSCE# is forced HIGH. Once the data register contains data and the interface is enabled, LSCE# is pulled LOW and remain LOW for the duration of the transmission.

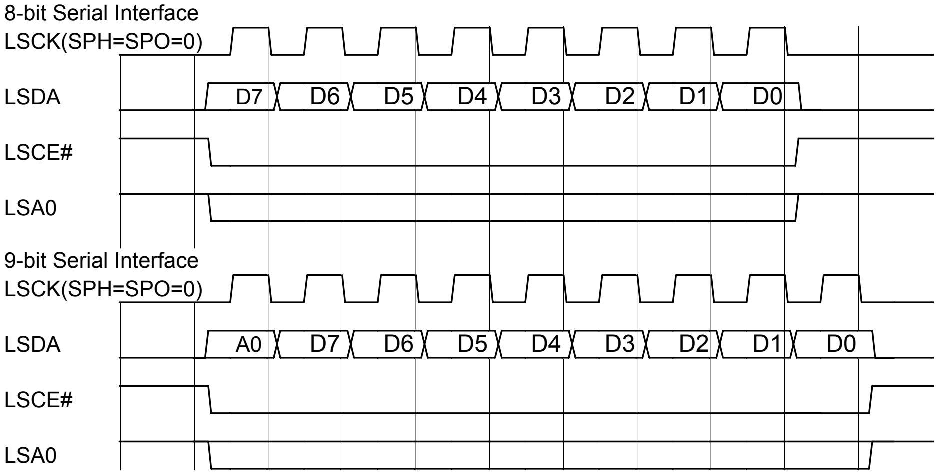


Figure 139 ***LCD Interface Transfer Timing Diagram***

LCD = 0x8012_0000

Address	Register Function	Width	Acronym
LCD + 0000h	LCD Interface Status Register	16	**LCD_STA**
LCD + 0004h	LCD Interface Interrupt Enable Register	16	**LCD_INTEN**
LCD + 0008h	LCD Interface Interrupt Status Register	16	**LCD_INTSTA**
LCD + 000ch	LCD Interface Frame Transfer Register	16	**LCD_START**
LCD + 0010h	LCD Parallel/Serial LCM Reset Register	16	**LCD_RSTB**
LCD + 0014h	LCD Serial Interface Configuration Register	16	**LCD_SCNF**
LCD + 0018h	LCD Parallel Interface 0 Configuration Register	32	**LCD_PCNF0**
LCD + 001ch	LCD Parallel Interface 1 Configuration Register	32	**LCD_PCNF1**
LCD + 0020h	LCD Parallel Interface 2 Configuration Register	32	**LCD_PCNF2**
LCD + 0024h	LCD Tearing Control Register	16	**LCD_TECON**
LCD + 0028h	LCD Parallel Interface Data Width Configuration Register	32	**LCD_PCNFDW**
LCD + 0030h	LCD ROI Window Write to Memory Address Register 0	32	**LCD_WROI_W2MADD0**
LCD + 0034h	LCD ROI Window Write to Memory Address Register 1	32	**LCD_WROI_W2MADD1**
LCD + 0038h	LCD ROI Window Write to Memory Address Register 2	32	**LCD_WROI_W2MADD2**
LCD + 0040h	LCD Main Window Size Register	32	**LCD_MWINSIZE**
LCD + 0044h	LCD ROI Window Write to Memory Offset Register	32	**LCD_WROI_W2MOFS**
LCD + 0048h	LCD ROI Window Write to Memory Control Register	16	**LCD_WROI_W2MCON**
LCD + 004Ch	Region of Interest Window Format Register	16	**LCD_WROIFMT**
LCD + 0050h	LCD ROI Window Control Register	32	**LCD_WROICON**
LCD + 0054h	LCD ROI Window Offset Register	32	**LCD_WROIOFS**
LCD + 0058h	LCD ROI Window Command Start Address Register	16	**LCD_WROICADD**
LCD + 005ch	LCD ROI Window Data Start Address Register	16	**LCD_WROIDADD**
LCD + 0060h	LCD ROI Window Size Register	32	**LCD_WROISIZE**
LCD + 0064h	LCD ROI Window Hardware Refresh Register	32	**LCD_WROI_HWREF**
LCD + 0068h	LCD ROI Direct Couple Register	32	**LCD_WROI_DC**
LCD + 006Ch	LCD ROI Window Background Color Register	32	**LCD_WROI_BGCLR**
LCD + 0070h	LCD Layer 0 Window Control Register	32	**LCD_L0WINCON**
LCD + 0074h	LCD Layer 0 Source Color Key Register	32	**LCD_L0WINSKEY**
LCD + 0078h	LCD Layer 0 Window Display Offset Register	32	**LCD_L0WINOFS**
LCD + 007ch	LCD Layer 0 Window Display Start Address Register	32	**LCD_L0WINADD**
LCD + 0080h	LCD Layer 0 Window Size	32	**LCD_L0WINSIZE**

LCD + 0084h	LCD Layer 0 Scroll Start Offset	32	**LCD_L0WINSCRL**
LCD + 0090h	LCD Layer 1 Window Control Register	32	**LCD_L1WINCON**
LCD + 0094h	LCD Layer 1 Source Color Key Register	32	**LCD_L1WINSKEY**
LCD + 0098h	LCD Layer 1 Window Display Offset Register	32	**LCD_L1WINOFS**
LCD + 009ch	LCD Layer 1 Window Display Start Address Register	32	**LCD_L1WINADD**
LCD + 00a0h	LCD Layer 1 Window Size	32	**LCD_L1WINSIZE**
LCD + 00a4h	LCD Layer 1 Scroll Start Offset	32	**LCD_L1WINSCRL**
LCD + 00b0h	LCD Layer 2 Window Control Register	32	**LCD_L2WINCON**
LCD + 00b4h	LCD Layer 2 Source Color Key Register	32	**LCD_L2WINSKEY**
LCD + 00b8h	LCD Layer 2 Window Display Offset Register	32	**LCD_L2WINOFS**
LCD + 00bch	LCD Layer 2 Window Display Start Address Register	32	**LCD_L2WINADD**
LCD + 00c0h	LCD Layer 2 Window Size	32	**LCD_L2WINSIZE**
LCD + 00c4h	LCD Layer 2 Scroll Start Offset	32	**LCD_L2WINSCRL**
LCD + 00d0h	LCD Layer 3 Window Control Register	32	**LCD_L3WINCON**
LCD + 00d4h	LCD Layer 3 Source Color Key Register	32	**LCD_L3WINSKEY**
LCD + 00d8h	LCD Layer 3 Window Display Offset Register	32	**LCD_L3WINOFS**
LCD + 00dch	LCD Layer 3 Window Display Start Address Register	32	**LCD_L3WINADD**
LCD + 00e0h	LCD Layer 3 Window Size	32	**LCD_L3WINSIZE**
LCD + 00e4h	LCD Layer 3 Scroll Start Offset	32	**LCD_L3WINSCRL**
LCD + 00f0h	LCD Layer 4 Window Control Register	32	**LCD_L4WINCON**
LCD + 00f4h	LCD Layer 4 Source Color Key Register	32	**LCD_L4WINSKEY**
LCD + 00f8h	LCD Layer 4 Window Display Offset Register	32	**LCD_L4WINOFS**
LCD + 00fch	LCD Layer 4 Window Display Start Address Register	32	**LCD_L4WINADD**
LCD + 0100h	LCD Layer 4 Window Size	32	**LCD_L4WINSIZE**
LCD + 0104h	LCD Layer 4 Scroll Start Offset	32	**LCD_L4WINSCRL**
LCD + 0110h	LCD Layer 5 Window Control Register	32	**LCD_L5WINCON**
LCD + 0114h	LCD Layer 5 Source Color Key Register	32	**LCD_L5WINSKEY**
LCD + 0118h	LCD Layer 5 Window Display Offset Register	32	**LCD_L5WINOFS**
LCD + 011ch	LCD Layer 5 Window Display Start Address Register	32	**LCD_L5WINADD**
LCD + 0120h	LCD Layer 5 Window Size	32	**LCD_L5WINSIZE**
LCD + 0124h	LCD Layer 5 Scroll Start Offset	32	**LCD_L5WINSCRL**
LCD + 0130h	LCD Color Management Coefficient Row 0	32	**LCD_CM_COEF_ROW0**
LCD + 0134h	LCD Color Management Coefficient Row 1	32	**LCD_CM_COEF_ROW1**
LCD + 0138h	LCD Color Management Coefficient Row 2	32	**LCD_CM_COEF_ROW2**
LCD + 0140h	LCD Piece-Wise Linear Gamma Segment 0~20	32	**LCD_GMA0~20**

© 2009 MediaTek Inc.

~0193h			
LCD + 0200h	Frame Buffer Compression Target Rate	32	**FBCE_TARGET_RATE**
LCD + 0204h	Frame Buffer Compression Input Mode	2	**FBCE_INPUT_MODE**
LCD + 0208h	Frame Buffer Compression Start	1	**FBCE_ACT_EN**
LCD + 0210h	Frame Buffer Compression Status	5	**FBCE_STATUS**
LCD + 0220h	Frame Buffer Decompression Control Register		**FBCD_CON**
LCD + 4000h	LCD Parallel Interface 0 Data	32	**LCD_PDAT0**
LCD + 4100h	LCD Parallel Interface 0 Command	32	**LCD_PCMD0**
LCD + 5000h	LCD Parallel Interface 1 Data	32	**LCD_PDAT1**
LCD + 5100h	LCD Parallel Interface 1 Command	32	**LCD_PCMD1**
LCD + 6000h	LCD Parallel Interface 2 Data	32	**LCD_PDAT2**
LCD + 6100h	LCD Parallel Interface 2 Command	32	**LCD_PCMD2**
LCD + 8000h	LCD Serial Interface 1 Data	16	**LCD_SDAT1**
LCD + 8100h	LCD Serial Interface 1 Command	16	**LCD_SCMD1**
LCD + 9000h	LCD Serial Interface 0 Data	16	**LCD_SDAT0**
LCD + 9100h	LCD Serial Interface 0 Command	16	**LCD_SCMD0**
LCD + C000h ~ CFFCh	LCD Gamma Correction LUT Register	32	**LCD_GAMMA**
LCD + D000h ~ D3FCh	LCD Color Palette LUT Register	32	**LCD_PAL**
LCD + d400h ~ D47Ch	LCD Interface Command/Parameter0 Register	32	**LCD_COMD0**
LCD + d480h ~ D4FCh	LCD Interface Command/Parameter1 Register	32	**LCD_COMD1**

Table 115 ***Memory map of LCD Interface***

5.20.2 Register Definitions

LCD +0000h LCD Interface Status Register LCD_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**TE_PEND**	**CMD_CPEND**	**DATA_PEND**	**RUN**
Type													R	R	R	R
Reset													0	0	0	0

RUN LCD Interface Running Status
DATA_PEND Data Pending Indicator in Hardware Trigger Mode
CMD_PEND Command Pending Indicator in Hardware Triggered Refresh Mode
TE_PEND Waiting Tearing Effect signal

LCD +0004h LCD Interface Interrupt Enable Register LCD_INTEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name												**FBCD_ERR**	**TE_DET**	**CMD_CPL**	**DATA_CPL**	**CPL**
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

CPL LCD Frame Transfer Complete Interrupt Control
DATA_CPL Data Transfer Complete in Hardware Triggered Refresh Mode Interrupt Control
CMD_CPL Command Transfer Complete in Hardware Trigger Refresh Mode Interrupt Control
TE_DET Tearing Effect signal detection interrupt control
FBCD_ERR Frame buffer decompression error interrupt control

LCD +0008h LCD Interface Interrupt Status Register LCD_INTSTA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**FBCD_ERR**	**TE_DET**	**CMD_CPL**	**DATA_CPL**	**CPL**
Type												R/W	R	R	R	R
Reset												0	0	0	0	0

CPL LCD Frame Transfer Complete Interrupt
DATA_CPL Data Transfer Complete in Hardware Triggered Refresh Mode Interrupt
CMD_CPL Command Transfer Complete in Hardware Triggered Refresh Mode Interrupt
TE_DET Tearing Effect signal detection interrupt
FBCD_ERR Frame buffer decompression error interrupt

LCD +000Ch LCD Interface Frame Transfer Register LCD_START

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**START**															
Type	R/W															
Reset	0															

START Start Control of LCD Frame Transfer

LCD +0010h LCD Parallel/Serial Interface Reset Register LCD_RSTB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RSTB**
Type																R/W
Reset																1

RSTB Parallel/Serial LCD Module Reset Control

LCD +0014h LCD Serial Interface Configuration Register LCD _SCNF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**52M**	**26M**	**GAMMA_ID**			**NON_DBI**	**CSP1**	**CSP0**	**CMR**	**CMG**	**CMB**	**8/9**	**DIV**		**SPH**	**SPO**
Type	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Type	0	0	11			0	0	0	0	0	0	0	0		0	0

SPO Clock Polarity Control
SPH Clock Phase Control
DIV Serial Clock Divide Select Bits

© 2009 MediaTek Inc.

0 reserved
1 engine clock/4
2 engine clock/8
3 engine clock/16

8/9 8-bit or 9-bit Interface Selection
CMB Color management for BLUE component.
CMG Color management for GREEN component.
CMR Color management for RED component.
CSP0 Serial Interface Chip Select 0 Polarity Control
CSP1 Serial Interface Chip Select 1 Polarity Control
NON_DBI non-DBI protocol serial interface

0 DBI type C serial interface, 8 or 9 bits per transaction
1 non-DBI type C serial interface, 8/9/16/18/24/32 bits per transaction
If the transmission source is layer update or command queue, the bits per transaction is defined in LCD_FMT[8:6]. If the transmission source is from AHB bus write command, the bits per transaction is defined by the data width of this AHB write command. If "8/9" bit is set, the first bit of transmission depends on the address of this write. If the write address is LCD_SDAT1 or LCD_SDAT0, the first bit is 0. If the write address is LCD_SCMD1 or LCD_SCMD0, the first bit is 1.

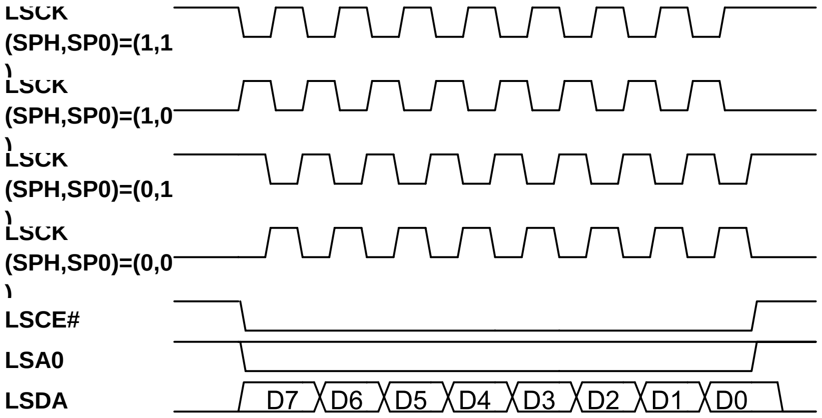
GAMMA_ID Gamma correction LUT ID

00 table 0
01 table 1
10 table 2
11 no table selected

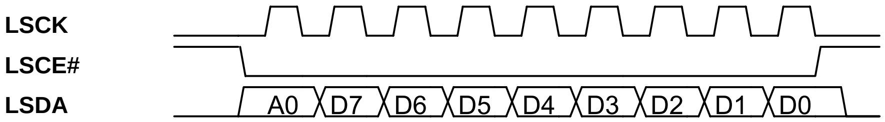
26M Enable 26MHz clock gating.
52M Enable 52MHz clock gating.

 © 2009 MediaTek Inc.

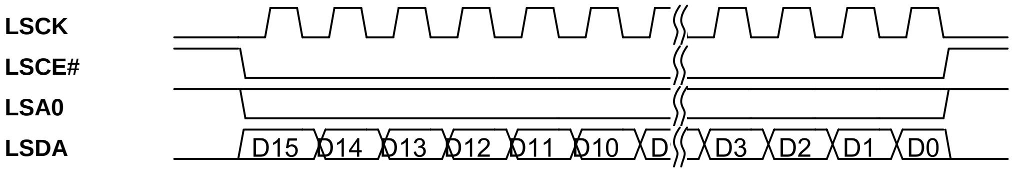
8bit (8/9=0, NON_DBI=0)



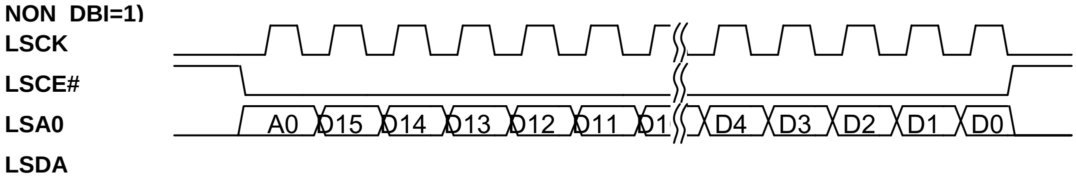
9bit (8/9=1, NON_DBI=0)



16bit (8/9=0, NON_DBI=1)



16bit+A0 (8/9=1, NON_DBI=1)



LCD +0018h LCD Parallel Interface Configuration Register 0 LCD_PCNF0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**C2WS**		**C2WH**		**C2RS**				**GAMMA_ID_R**		**GAMMA_ID_G**		**GAMMA_ID_B**		**DW**	
Type	R/W		R/W		R/W				R/W		R/W		R/W			
	0		0		0				11		11		11			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**52M**	**26M**		**WST**					**CMR**	**CMG**	**CMB**	**RLT**				
Type	R/W	R/W		R/W					R/W	R/W	R/W	R/W				
Reset	0	0		0					0	0	0	0				

RLT Read Latency Time
CMB Color management for BLUE component.
CMG Color management for GREEN component.
CMR Color management for RED component.
WST Write Wait State Time
26M Enable 26MHz clock gating.
52M Enable 52MHz clock gating.
DW Move to LCD_PCNFDW LCD+0028h
GAMMA_ID _R Gamma Correction LUT ID for Red Component
- **00** table 0
- **01** table 1
- 10 table 2
- 11 no table selected

GAMMA_ID_G Gamma correction LUT ID for Green Component
- **00** table 0
- **01** table 1
- **10** table 2
- **11** no table selected

GAMMA_ID_B Gamma correction LUT ID for Blue Component
- **00** table 0
- **01** table 1
- **10** table 2
- **11** no table selected

C2RS Chip Select (LPCE#) to Read Strobe (LRD#) Setup Time
C2WH Chip Select (LPCE#) to Write Strobe (LWR#) Hold Time
C2WS Chip Select (LPCE#) to Write Strobe (LWR#) Setup Time

LCD +001Ch LCD Parallel Interface Configuration Register 1 LCD_PCNF1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**C2WS**		**C2WH**		**C2RS**						**GAMMA_ID**				**DW**	
Type	R/W		R/W		R/W						R/W					
	0		0		0						11					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**52M**	**26M**		**WST**					**CMR**	**CMG**	**CMB**	**RLT**				
Type	R/W	R/W		R/W					R/W	R/W	R/W	R/W				
Reset	0	0		0					0	0	0	0				

RLT Read Latency Time
CMB Color management for BLUE component.
CMG Color management for GREEN component.
CMR Color management for RED component.
WST Write Wait State Time
26M Enable 26MHz clock gating.
52M Enable 52MHz clock gating.

DW Move to LCD_PCNFDW LCD+0028h
GAMMA_ID Gamma correction LUT ID
- **00** table 0
- **01** table 1
- **10** table 2
- **11** no table selected

C2RS Chip Select (LPCE#) to Read Strobe (LRD#) Setup Time
C2WH Chip Select (LPCE#) to Write Strobe (LWR#) Hold Time
C2WS Chip Select (LPCE#) to Write Strobe (LWR#) Setup Time

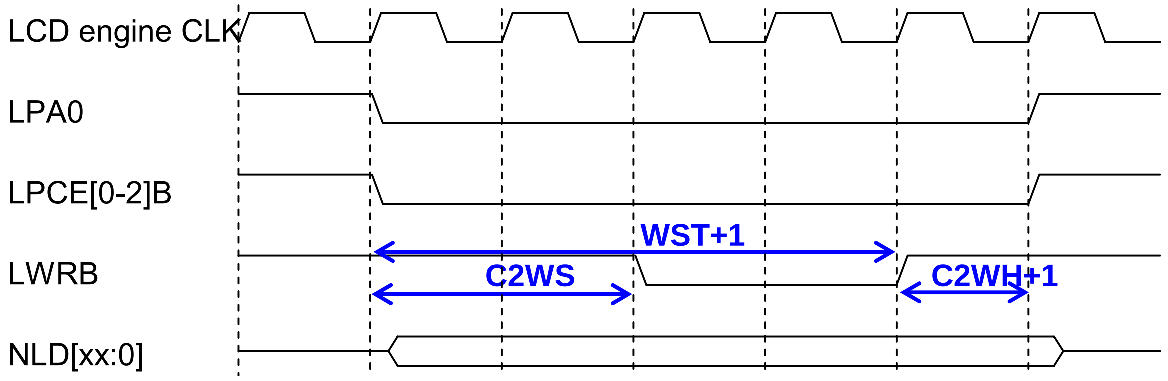
LCD +0020h LCD Parallel Interface Configuration Register 2 LCD_PCNF2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	C2WS		C2WH		C2RS						GAMMA_ID				DW	
Type	R/W		R/W		R/W						R/W					
	0		0		0						11					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	52M	26M		WST					CMR	CMG	CMB	RLT				
Type	R/W	R/W		R/W					R/W	R/W	R/W	R/W				
Reset	0	0		0					0	0	0	0				

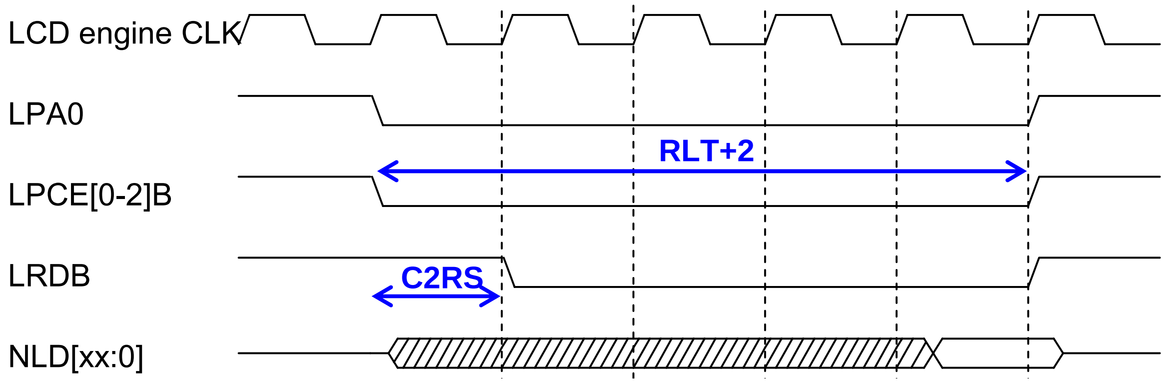
RLT Read Latency Time
CMB Color management for BLUE component.
CMG Color management for GREEN component.
CMR Color management for RED component.
WST Write Wait State Time
26M Enable 26MHz clock gating.
52M Enable 52MHz clock gating.
DW Move to LCD_PCNFDW LCD+0028h
GAMMA_ID Gamma correction LUT ID
- **00** table 0
- **01** table 1
- **10** table 2
- **11** no table selected

C2RS Chip Select (LPCE#) to Read Strobe (LRD#) Setup Time
C2WH Chip Select (LPCE#) to Write Strobe (LWR#) Hold Time
C2WS Chip Select (LPCE#) to Write Strobe (LWR#) Setup Time

Paralle Interface Write timing
C2WS=2, WST=3, C2WH=0, C2WS must <= WST



Read timing
C2RS=1, RLT=3



LCD Controller Synchronization Modes

When TE_EN is enabled, LCD controller will synchronize its updating to LCM refresh timing. And it supports two synchronizing modes depending on TE_MODE.

TE_MODE value	Synchronization Mode
0	Vertical Synchronization mode. LCD controller starts to update LCM when it detects a TE signal.
1	Vertical and Horizontal Synchronization mode. LCD controller starts to update LCM when it detects a vertical TE and following (TE_HS_CNT_MATCH_VALUE+1) horizontal TEs.

Table 116 ***LCD Controller Synchronization Mode***

TE Signal Polarity

TE_EDGE_SEL can be used to select TE polarity for TE signal detection.

TE_EDGE_SEL value	TE signal detection
0	Detect a TE signal at its rising edge. This setting is for active high TE signal.
1	Detect a TE signal at its falling edge. This setting if for active low TE signal.

Table 117 ***TE Signal Polarity***

Vertical Synchronization Mode

In Vertical Synchronization mode, the LCD controller assumes TE input is only a vertical synchronization signal.

In this mode, LCD controller starts to update LCM after each rising edge of the TE input (or falling edge, if TE_EDGE_SEL is set to 1) (see Figure 3).

In this mode, TE_VS_WIDTH_LIMIT, TE_VS_WIDTH_CNT_DIV, TE_HS_CNT_MATCH_VALUE are not used.

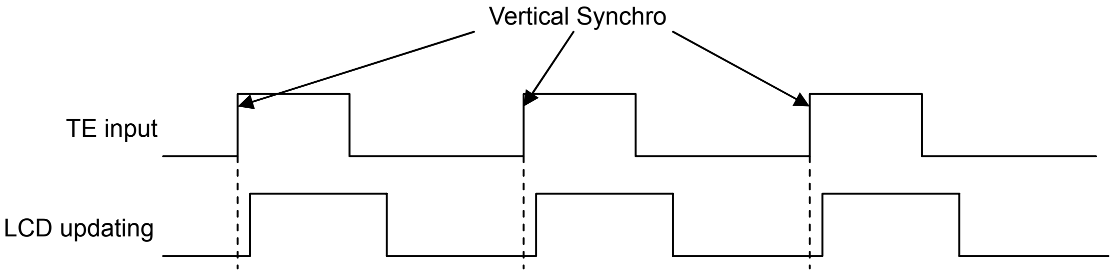


Figure 140 ***Vertical Synchronization Mode***

Vertical and Horizontal Synchronization Mode

In Vertical and Horizontal Synchronization mode, the LCD controller assumes TE input is an OR of vertical and horizontal synchronization signal.

TE_VS_WIDTH_LIMIT gives the minimum time that TE input must stay active to be detected by the LCD controller as a vertical synchronization. This time is (TE_VS_WIDTH_LIMIT+1)*(divisor specified by TE_VS_WIDTH_CNT_DIV) clock cycles. Any pulse longer than this time is considered a vertical synchronization. Any pulse shorter than this time is considered a horizontal synchronization.

Once a vertical synchronization has been detected, the LCD controller counts the active edges on the TE input.

When the number of active edges reaches TE_HS_CNT_MATCH_VALUE, LCD controller starts to update LCM. If a new vertical synchronization is detected before HS_CNT_MATCH_VALUE horizontal synchro have been detected, the counter is reset and horizontal synchronization count begins once again (see Figure 4).

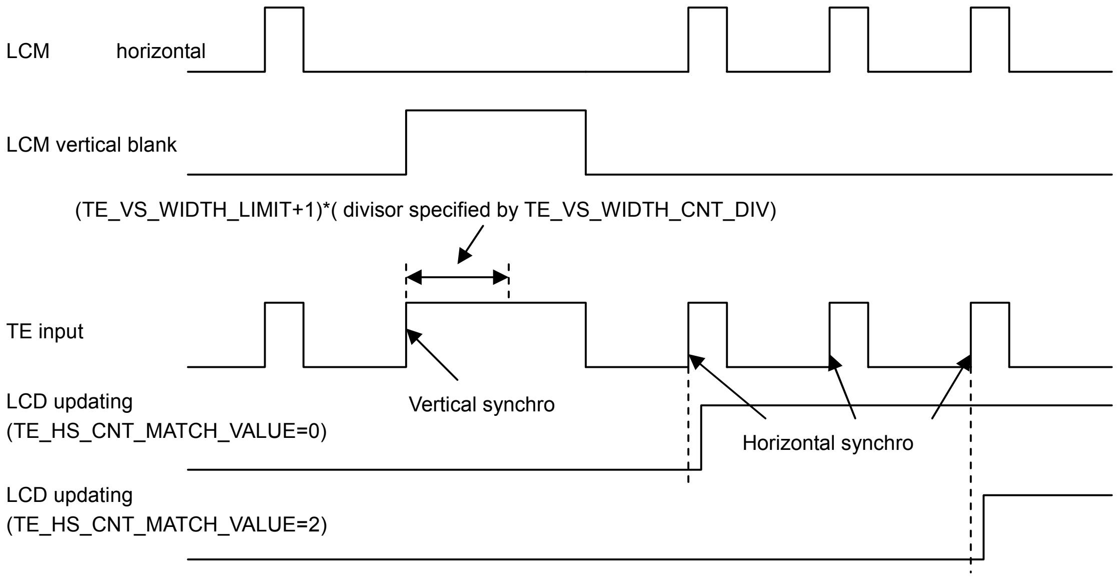


Figure 141 ***Vertical and Horizontal Synchronization Mode***

LCD +0024h LCD Tearing Control Register LCD_TECON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TE_VS_WIDTH_LIMIT														TE_VS_WIDTH_CNT_DIV	
Type	R/W														R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		TE_HS_CNT_MATCH_VALUE											TE_REPEAT	TE_MODE	TE_EDGE_SEL	TE_EN
Type		R/W											R/W	R/W	R/W	R/W
Reset		NA											0	0	0	0

TE_EN Enable tearing control. LCD controller will synchronize to LCM refresh timing.

TE_EDGE_SEL Select sync edge.

- **0** Rising edge
- **1** Falling edge

TE_MODE Tearing control mode.

- **0** Start transmission to LCD after TE edge.
- **1** Wait (TE_HS_CNT_MATCH_VALUE+1) lines after vertical TE edge to start transmission to LCD.

TE_REPEAT Repeat mode.

- **0** update LCM once every TE signal coming.
- **1** repeat updating LCM after TE signal coming.

TE_HS_CNT_MATCH_VALUE Trigger LCD update after delaying (TE_HS_CNT_MATCH_VALUE+1) lines after vertical TE edge.

TE_VS_WIDTH_CNT_DIV Engine clock divisor for vertical TE detection.

- **00** divide engine clock by 8
- **01** divide engine clock by 16

10 divide engine clock by 32

11 divide engine clock by 64

TE_VS_WIDTH_LIMIT If the width of a TE pulse is larger than (TE_VS_WIDTH_LIMIT+1)* (divisor specified by TE_VS_WIDTH_CNT_DIV), it is a vertical TE, otherwise it is a horizontal TE.

LCD +0028h LCD Parallel Interface Data Width Configuration Register LCD_PCNFDW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														**PCNF2_DW**		
Type														R/W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**PCNF1_DW**								**PCNF0_DW**		
Type						R/W								R/W		

PCNF0_DW Data width of parallel interface 0

000 8 bit

001 9 bit

010 16 bit

011 18 bit

100 24 bit

others reserved

PCNF1_DW Data width of parallel interface 1, bit encode is the same as PCNF0_DW

PCNF2_DW Data width of parallel interface 2, bit encode is the same as PCNF0_DW

LCD +4000h LCD Parallel 0 Interface Data LCD_PDAT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DATA[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA[15:0]**															
Type	R/W															

DATA Writing to LCD+4000 will drive LPA0 low when sending this data out in parallel BANK0, while writing to LCD+4100 will drive LPA0 high.

LCD +5000h LCD Parallel 1 Interface Data LCD_PDAT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DATA[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA[15:0]**															
Type	R/W															

DATA Writing to LCD+5000 will drive LPA1 low when sending this data out in parallel BANK1, while writing to LCD+5100 will drive LPA1 high

LCD +6000h LCD Parallel 2 Interface Data LCD_PDAT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	R/W															

DATA Writing to LCD+6000 will drive LPA2 low when sending this data out in parallel BANK2, while writing to LCD+6100 will drive LPA2 high

LCD +8000/8100h LCD Serial Interface 1 Data LCD_SDAT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA							
Type									W							

DATA Writing to LCD+8000 will drive LSA0 low while sending this data out in serial BANK1, while writing to LCD+8100 will drive LSA0 high

LCD +9000/9100h LCD Serial Interface 0 Data LCD_SDAT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA							
Type									W							

DATA Writing to LCD+9000 will drive LSA0 low while sending this data out in serial BANK0, while writing to LCD+9100 will drive LSA0 high

LCD +0030h Region of Interest Window Write to Memory Address FB0 Register LCD_WROI_W2MADD0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W2M_ADDR0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W2M_ADDR0															
Type	R/W															

W2M_ADDR0 Write to memory address for Frame Buffer 0. This address must be 8byte-aligned.

LCD +0034h Region of Interest Window Write to Memory Address FB1 Register LCD_WROI_W2MADD1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W2M_ADDR1															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W2M_ADDR1															
Type	R/W															

W2M_ADDR1 Write to memory address for Frame Buffer 1. This address must be 8byte-aligned.

LCD +0038h Region of Interest Window Write to Memory Address FB2 Register LCD_WROI_W2MADD2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W2M_ADDR2															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W2M_ADDR2															
Type	R/W															

W2M_ADDR2 Write to memory address for Frame Buffer 2. This address must be 8byte-aligned.

LCD +0040h Main Window Size Register LCD_MWINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							ROW									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							COLUMN									
Type							R/W									

COLUMN 10-bit Virtual Image Window Column Size

ROW 10-bit Virtual Image Window Row Size

LCD +0044h Region of Interest Window Write to Memory Offset Register LCD_WROI_W2MOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

This control register is used to specify the offset of the ROI window from the LCD_WROI_W2MADDR when writing the ROI window's content to memory.

X-OFFSET the x offset of ROI window in the destination memory.

Y-OFFSET the y offset of ROI window in the destination memory.

LCD +0048h Region of Interest Window Write to Memory Control Register LCD_WROI_W2MCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									WB_DIS			DLY_EN	FBSEQ_RST		FB2_EN	FB1_EN
Type									R/W			R/W	R/W		R/W	R/W
Reset									0			0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OUTPUT_ALPHA										DC_OUT_EN	ADDINC_DISABLE	DISCON	W2M_FORMAT		W2LCM
Type	R/W										R/W	R/W	R/W	R/W		R/W
Reset	0xff										0	0	0	0		0

This control register is effective only when the W2M bit is set in LCD_WROICON register.

W2LCM Write to LCM simultaneously.

W2M_FORMAT Write to memory format.

00 RGB565

01 RGB888

10 ARGB8888

DISCON Block Write Enable Control. By setting both DISCON and W2M to 1, the LCD controller will write out the ROI pixel data as a part of MAIN window, using the width of MAIN window to calculate the write-out address. If this bit is not set, the ROI window will be written to memory in continuous addresses.

ADDINC_DISABLE Disable address increase when writing to memory.

DC_OUT_EN Enable direct couple to rotator 3.

OUTPUT_ALPHA Output Alpha value.

FB1_EN Enable frame buffer 1, which starting address is LCD_WROI_W2MADD1.

FB2_EN Enable frame buffer 2, which starting address is LCD_WROI_W2MADD2.

DLY_EN Enable that LCD overlay engine start to write to memory after DPI indication.

WB_DIS Disable LCD dedicated write buffer. The write buffer is used to enhance the memory write efficiency.

FBSEQ_RST Reset the next update frame buffer to FB0. This bit will remain 1 after set until it takes effect.

LCD +004Ch Region of Interest Window Format Register LCD_WROIFMT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DSI SWP	EN_BLS	WR2DSI				FORMAT								
Type		R/W	R/W	R/W				R/W								

FORMAT LCD Module Data Format

Bit 0 : in BGR sequence, otherwise in RGB sequence.

Bit 1 : LSB first, otherwise MSB first.

Bit 2 : padding bits on MSBs, otherwise on LSBs.

Bit 5-3 : 000 for RGB332, 001 for RGB444, 010 for RGB565, 011 for RGB666, 100 for RGB888.

Bit 8-6 : 000 for 8-bit interface, 001 for 16-bit interface, 010 for 9-bit interface, 011 for 18-bit interface, 100 for 24-bit interface, others for reserved.

Note: When the interface is configured as 9 bit or 18 bit, the field of bit5-2 is ignored.

WR2DSI Write to DSI.

EN_BLS Enable back light scaling.

DSI_SWP Swap data order for DSI transmission. 1: low byte first, 0: high byte first

00000000	8bit	1cycle/1pixel	RGB3.3.2	RRRGGGBB
00000001		1cycle/1pixel	RGB3.3.2	BBGGGRRR
00001000		3cycle/2pixel	RGB4.4.4	RRRRGGGG BBBBRRRR

				GGGGBBBB
00001011		3cycle/2pixel	RGB4.4.4	GGGGRRRR RRRRBBBB BBBBGGGG
00010000		2cycle/1pixel	RGB5.6.5	RRRRRGGG GGGBBBBB
00010011		2cycle/1pixel	RGB5.6.5	GGGRRRRR BBBBBGGG
00011000		3cycle/1pixel	RGB6.6.6	RRRRRRXX GGGGGGXX BBBBBBXX
00011100		3cycle/1pixel	RGB6.6.6	XXRRRRRR XXGGGGGG XXBBBBBB
00100000		3cycle/1pixel	RGB8.8.8	RRRRRRRR GGGGGGGG BBBBBBBB
10xxxx00	9bit	2cycle/1pixel	RGB6.6.6	RRRRRRGGG GGGBBBBBB
10xxxx11		2cycle/1pixel	RGB6.6.6	GGGRRRRRR BBBBBBGGG
01000000	16bit	1cycle/2pixel	RGB3.3.2	RRRGGGBBRRRGGGBB
01000010		1cycle/2pixel	RGB3.3.2	RRRGGGBBRRRGGGBB
01000001		1cycle/2pixel	RGB3.3.2	BBGGGRRRBBGGGRRR
01000011		1cycle/2pixel	RGB3.3.2	BBGGGRRRBBGGGRRR
01001100		1cycle/1pixel	RGB4.4.4	XXXXRRRRGGGGBBBB
01001101		1cycle/1pixel	RGB4.4.4	XXXXBBBBGGGGRRRR
01001000		1cycle/1pixel	RGB4.4.4	RRRRGGGGBBBBXXXX
01001001		1cycle/1pixel	RGB4.4.4	BBBBGGGGRRRRXXXX
01010000		1cycle/1pixel	RGB5.6.5	RRRRRGGGGGGBBBBB
01010001		1cycle/1pixel	RGB5.6.5	BBBBBGGGGGGRRRRR
01011100		3cycle/2pixel	RGB6.6.6	XXXXRRRRRRGGGGGG XXXXBBBBBBRRRRRR XXXXGGGGGGBBBBBB
01011111		3cycle/2pixel	RGB6.6.6	XXXXGGGGGGRRRRRR XXXXRRRRRRBBBBBB XXXXBBBBBBGGGGGG
01011000		3cycle/2pixel	RGB6.6.6	RRRRRRGGGGGGXXXX BBBBBBRRRRRRXXXX GGGGGGBBBBBBXXXX
01011011		3cycle/2pixel	RGB6.6.6	GGGGGGRRRRRRXXXX RRRRRRBBBBBBXXXX

				BBBBBBGGGGGGXXXX
01100000		3cycle/2pixel	RGB8.8.8	RRRRRRRRGGGGGGGG BBBBBBBBRRRRRRRR GGGGGGGGBBBBBBBB
01100011		3cycle/2pixel	RGB8.8.8	GGGGGGGGRRRRRRRR RRRRRRRRBBBBBBBB BBBBBBBBRRRRRRRR
11xxxx00	18bit	1cycle/1pixel	RGB6.6.6	RRRRRRGGGGGGBBBBBB
11xxxx01		1cycle/1pixel	RGB6.6.6	BBBBBBGGGGGGRRRRRR
11100000		3cycle/2pixel	RGB8.8.8	RRRRRRRRGGGGGGGG BBBBBBBBRRRRRRRR GGGGGGGGBBBBBBBB
11100011		3cycle/2pixel	RGB8.8.8	GGGGGGGGRRRRRRRR RRRRRRRRBBBBBBBB BBBBBBBBRRRRRRRR

LCD +0050h Region of Interest Window Control Register LCD_WROICON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EN0	EN1	EN2	EN3	EN4	EN5	PERIOD									
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENC	W2M	COM_SEL	COMMAND					FORMAT							
Type	R/W	R/W	R/W	R/W												

FORMAT Move to LCD_WROIFMT LCD+004Ch

COM_SEL Command Queue ID Selection

COMMAND Number of Commands to be sent to LCD module. Maximum is 31.

W2M Enable Write to Memory

ENC Command Transfer Enable Control

PERIOD Waiting period between two consecutive transfers, effective for both data and command.

ENn Layer Window Enable Control

LCD +0054h Region of Interest Window Offset Register LCD_WROIOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

X-OFFSET ROI Window Column Offset

Y-OFFSET ROI Window Row Offset

LCD +0058h Region of Interest Window Command Start Address Register LCD_WROICADD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															

Type	R/W

ADDR ROI Window Command Address. Only writing to LCD modules is allowed.

LCD +005Ch Region of Interest Window Data Start Address Register LCD_WROIDADD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR ROI Window Data Address Only writing to LCD modules is allowed.

LCD +0060h Region of Interest Window Size Register LCD_WROISIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

COLUMN ROI Window Column Size

ROW ROI Window Row Size

LCD +0064h Region of Interest Window Hardware Refresh Register LCD_WROI_HWREF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EN0**	**EN1**	**EN2**	**EN3**	**EN4**	**EN5**									**HWREF_SEL**	
Type	R/W	R/W	R/W	R/W	R/W	R/W									R/W	
Reset	0	0	0	0	0	0									0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**HWEN**							**HWREF**
Type									R/W							R/W
Reset									0							0

EN*n* Enable layer n source address from Image_DMA.

HWEN Enable hardware triggered LCD fresh.

HWREF_SEL Select hardware triggered source.

- **00** triggered by IRT1.
- **01** triggered by IBW1.
- **10** triggered by IRT2 (without base address).
- **11** triggered by IBW2 (without base address).

HWREF Starting the hardware triggered LCD frame transfer.

LCD +0068h Region of Interest Window Direct Couple Register LCD_WROI_DC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EN0**	**EN1**	**EN2**	**EN3**	**EN4**	**EN5**										
Type	R/W	R/W	R/W	R/W	R/W	R/W										
Reset	0	0	0	0	0	0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

© 2009 MediaTek Inc.

Name	DC_SEL0	DC_SEL1	DC_SEL2	DC_SEL3	DC_SEL4	DC_SEL5		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

EN*n* Enable layer n source data from Image_DMA.

DC_SELn Select source layer n data.

- **00** Reserved.
- **01** IBW1
- **10** IRT2
- **11** IBW2

Note : When direct couple is enabled on mulitple layers, the source data of each layer should be different.

LCD +006Ch Region of Interest Background Color Register LCD_WROI_BGCLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RED[7:0]							
Type									R/W							
Reset									1111_1111							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GREEN[7:0]								BLUE[7:0]							
Type	R/W								R/W							
Reset	1111_1111								1111_1111							

RED Red component of ROI window's background color

GREEN Green component of ROI window's background color

BLUE Blue component of ROI window's background color

LCD +0070h Layer 0 Window Control Register LCD_L0WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												READ_CACHE_DIS		GMA_EN	SCRL_EN	SWP
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC	KEYEN	ROTATE			CLRDPT		OPAEN	OPA							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.

OPAEN Opacity enabled

CLRDPT Color format

- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration

- **000** 0 degree rotation

001 90 degree rotation counterclockwise
010 180 degree rotation counterclockwise
011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +0074h Layer 0 Source Color Key Register LCD_L0WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCKEY[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRCKEY[15:0]															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +0078h Layer 0 Window Display Offset Register LCD_L0WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 0 Window Row Offset
X-OFFSET Layer 0 Window Column Offset

LCD+007Ch Layer 0 Window Display Start Address Register LCD_L0WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR Layer 0 Window Data Address. Note that the layer start address must be 8byte-aligned.

LCD +0080h Layer 0 Window Size LCD_L0WINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

ROW Layer 0 Window Row Size

COLUMN Layer 0 Window Column Size

LCD +0084h Layer 0 Scroll Start Offset LCD_L0WINSCRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 0 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L0WINSIZE.ROW

X-OFFSET Layer 0 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L0WINSIZE.COLUMN

LCD +0090h Layer 1 Window Control Register LCD_L1WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**READ_CACHE_DIS**		**GMA_EN**	**SCRL_EN**	**SWP**
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC**	**KEYEN**	**ROTATE**			**CLRDPT**		**OPAEN**	**OPA**							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.

OPAEN Opacity enabled

CLRDPT Color format

- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration

- **000** 0 degree rotation
- **001** 90 degree rotation counterclockwise
- **010** 180 degree rotation counterclockwise

011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +0094h Layer 1 Source Color Key Register LCD_L1WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCKEY[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRCKEY[15:0]															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +0098h Layer 1 Window Display Offset Register LCD_L1WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 1 Window Row Offset
X-OFFSET Layer 1 Window Column Offset

LCD+009Ch Layer 1 Window Display Start Address Register LCD_L1WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR Layer 1 Window Data Address. Note that the layer start address must be 8byte-aligned.

LCD +00A0h Layer 1 Window Size LCD_L1WINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							ROW									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							COLUMN									
Type							R/W									

ROW Layer 1 Window Row Size
COLUMN Layer 1 Window Column Size

LCD +00A4h Layer 1 Scroll Start Offset LCD_L1WINSCRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 1 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L1WINSIZE.ROW
X-OFFSET Layer 1 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L1WINSIZE.COLUMN

LCD +00B0h Layer 2 Window Control Register LCD_L2WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												READ_CACHE_DIS		GMA_EN	SCRL_EN	SWP
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC	KEYEN	ROTATE			CLRDPT		OPAEN	OPA							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.
OPAEN Opacity enabled
CLRDPT Color format
- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration
- **000** 0 degree rotation
- **001** 90 degree rotation counterclockwise
- **010** 180 degree rotation counterclockwise

011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +00B4h Layer 2 Source Color Key Register LCD_L2WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SRCKEY[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRCKEY[15:0]**															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +00B8h Layer 2 Window Display Offset Register LCD_L2WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 2 Window Row Offset
X-OFFSET Layer 2 Window Column Offset

LCD+00BCh Layer 2 Window Display Start Address Register LCD_L2WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	R/W															

ADDR Layer 1 Window Data Address Note that the layer start address must be 8byte-aligned.

LCD +00C0h Layer 2 Window Size — LCD_L2WINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

ROW Layer 2 Window Row Size

COLUMN Layer 2 Window Column Size

LCD +00C4h Layer 2 Scroll Start Offset — LCD_L2WINSCRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 2 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L2WINSIZE.ROW

X-OFFSET Layer 2 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L2WINSIZE.COLUMN

LCD +00D0h Layer 3 Window Control Register — LCD_L3WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**READ_CACHE_DIS**		**GMA_EN**	**SCRL_EN**	**SWP**
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC**	**KEYEN**	**ROTATE**			**CLRDPT**		**OPAEN**	**OPA**							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.

OPAEN Opacity enabled

CLRDPT Color format

- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration

- **000** 0 degree rotation
- **001** 90 degree rotation counterclockwise
- **010** 180 degree rotation counterclockwise

© 2009 MediaTek Inc.

011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +00D4h Layer 3 Source Color Key Register LCD_L3WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCKEY[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRCKEY[15:0]															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +00D8h Layer 3 Window Display Offset Register LCD_L3WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 3 Window Row Offset
X-OFFSET Layer 3 Window Column Offset

LCD+00DCh Layer 3 Window Display Start Address Register LCD_L3WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR Layer 3 Window Data Address Note that the layer start address must be 8byte-aligned.

LCD +00E0h Layer 3 Window Size LCD_L3WINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

ROW Layer 3 Window Row Size

COLUMN Layer 3 Window Column Size

LCD +00E4h Layer 3 Scroll Start Offset LCD_L3WINSCRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 3 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L3WINSIZE.ROW

X-OFFSET Layer 3 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L3WINSIZE.COLUMN

LCD +00F0h Layer 4 Window Control Register LCD_L4WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**READ_CACHE_DIS**		**GMA_EN**	**SCRL_EN**	**SWP**
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC**	**KEYEN**	**ROTATE**			**CLRDPT**		**OPAEN**	**OPA**							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.

OPAEN Opacity enabled

CLRDPT Color format

- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration

- **000** 0 degree rotation
- **001** 90 degree rotation counterclockwise
- **010** 180 degree rotation counterclockwise

011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +00F4h Layer 4 Source Color Key Register LCD_L4WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCKEY[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRCKEY[15:0]															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +00F8h Layer 4 Window Display Offset Register LCD_L4WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 4 Window Row Offset
X-OFFSET Layer 4 Window Column Offset

LCD+00FCh Layer 4 Window Display Start Address Register LCD_L4WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR Layer 4 Window Data Address Note that the layer start address must be 8byte-aligned.

LCD +0100h Layer 4 Window Size — LCD_L4WINSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

ROW Layer 4 Window Row Size

COLUMN Layer 4 Window Column Size

LCD +0104h Layer 4 Scroll Start Offset — LCD_L4WINSCRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 4 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L4WINSIZE.ROW

X-OFFSET Layer 4 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L4WINSIZE.COLUMN

LCD +0110h Layer 5 Window Control Register — LCD_L5WINCON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**READ_CACHE_DIS**		**GMA_EN**	**SCRL_EN**	**SWP**
Type												R/W		R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC**	**KEYEN**	**ROTATE**			**CLRDPT**		**OPAEN**	**OPA**							
Type	R/W	R/W	R/W			R/W		R/W	R/W							

OPA Opacity value, used as constant alpha value.

OPAEN Opacity enabled

CLRDPT Color format

- **00** 8bpp indexed color.
- **01** RGB 565
- **10** ARGB 8888
- **11** RGB 888

ROTATE Rotation Configuration

- **000** 0 degree rotation
- **001** 90 degree rotation counterclockwise
- **010** 180 degree rotation counterclockwise

011 270 degree rotation counterclockwise
100 Horizontal flip
101 Horizontal flip then 90 degree rotation counterclockwise
110 Horizontal flip then 180 degree rotation counterclockwise
111 Horizontal flip then 270 degree rotation counterclockwise

KEYEN Source Key Enable Control
SRC Disable auto-increment of the source pixel address
SWP Swap high byte and low byte of pixel data
SCRL_EN Enable scroll effect
GMA_EN Enable piece-wise linear gamma correction
READ_CACHE_DIS Disable cache for this layer

LCD +0114h Layer 5 Source Color Key Register LCD_L5WINSKEY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCKEY[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRCKEY[15:0]															
Type	R/W															

SRCKEY Transparent color key of the source image.

LCD +0118h Layer 5 Window Display Offset Register LCD_L5WINOFS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Y-OFFSET									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							X-OFFSET									
Type							R/W									

Y-OFFSET Layer 5 Window Row Offset
X-OFFSET Layer 5 Window Column Offset

LCD+011Ch Layer 5 Window Display Start Address Register LCD_L5WINADD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR															
Type	R/W															

ADDR Layer 5 Window Data Address Note that the layer start address must be 8byte-aligned.

LCD +0120h Layer 5 Window Size **LCD_L5WINSIZE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**ROW**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**COLUMN**									
Type							R/W									

ROW Layer 5 Window Row Size

COLUMN Layer 5 Window Column Size

LCD +01284h Layer 5 Scroll Start Offset **LCD_L5WINSCRL**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**Y-OFFSET**									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**X-OFFSET**									
Type							R/W									

Y-OFFSET Layer 5 Scroll Start Row Offset, its value must satisfy Y-OFFSET < LCD_L5WINSIZE.ROW

X-OFFSET Layer 5 Scroll Start Column Offset, its value must satisfy X-OFFSET < LCD_L5WINSIZE.COLUMN

LCD+0130h Color Management Matrix Coefficient ROW0 **LCD_COEF_ROW0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**COEF_ROW0[29:16]**													
Type						R/W										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COEF_ROW0[15:0]**															
Type	R/W															

COEF_ROW0 Matrix row 0 for color management. Each coefficient is represented in signed 2.8 format (10bits).

LCD+0134h Color Management Matrix Coefficient ROW1 **LCD_COEF_ROW1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**COEF_ROW1[29:16]**													
Type						R/W										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COEF_ROW1[15:0]**															
Type	R/W															

COEF_ROW1 Matrix row 1 for color management. Each coefficient is represented as signed 2.8 format (10bits).

LCD+0138h Color Management Matrix Coefficient ROW2 LCD_COEF_ROW2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			COEF_ROW2[29:16]													
Type						R/W										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COEF_ROW2[15:0]															
Type	R/W															

COEF_ROW2 Matrix row 2 for color management. Each coefficient is represented as signed 2.8 format (10bits).

LCD+0140h ~ 0193h LCD piece-wise linear Gamma segment 0~19 LCD_GMA0~20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SLOPE0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YOFF0								XOFF0							
Type	R/W								R/W							

COEF_ROW2 Matrix row 2 for color management. Each coefficient is represented as signed 2.8 format (10bits).

LCD+0200h Frame buffer compression target rate FBCE_TARGET_RATE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FBCE_LINE_LENGTH											
Type					R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FBCE_TARGET_RATE				
Type												R/W				
Reset												0				

FBCE_TARGET_RATE range from 12~24 for RGB888 input mode, and range from 10~16 for RGB565 input mode. The compression rate is as FBCE_TARGET_RATE/24 for RGB888 input mode, and as FBCE_TARGET_RATE/16 for RGB565 input mode. This rate is that how many bytes will be encoded for a set (8 pixels). Ex. For RGB888, ratio=50%, then target rate = 24*50% = 12. RGB565, ratio=62.5%, then target rate = 16*62.5% = 10. Maxima compression rate is 50% for RGB888, and 62.5% for RGB565.

FBCE_LINE_LENGTH The bit stream length of a compressed line. Its value is calculated by the following equation.

$$FBCE_LINE_LENGTH = \lceil pixels_per_line/8 \rceil * FBCE_TARGET_RATE + 2$$

LCD+0204h Frame buffer compression input mode FBCE_INPUT_MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FBCE_PARTIAL	FBCE_INPUT_MODE	
Type														R/W	R/W	
Reset														0	0	

FBCE_INPUT_MODE

- **0** RGB888
- **1** RGB565
- **others** reserved

FBCE_PARTIAL FBCE partial update mode.

LCD+0208h Frame buffer compression start FBCE_ACT_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FBCE_ACT_EN
Type																R/W
Reset																0

FBCE_ACT_EN Frame buffer compression enable, it will start to encode after LCD_START is set.

LCD+0210h Frame Buffer Compression Status FBCE_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FBCD_ERR	FLAG2	FLAG1	FLAG0
Type													R	R	R	R

FLAG0 encode residual < 0.

FLAG1 encode flush with qfifo or efifo not zero.

FLAG2 frame buffer compression is running.

FBCD_ERR frame buffer decompression decoding error.

LCD+0220h Frame Buffer Decompression Target Rate FBCD_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EN0	EN1	EN2	EN3	EN4	EN5									FBCD_INPUT_MODE	
Type	R/W	R/W	R/W	R/W	R/W	R/W									R/W	
Reset	0	0	0	0	0	0									0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					FBCD_LINE_LENGTH											
Type					R/W											

FBCD_LINE_LENGTH The bit stream length of a compressed line. Its value is calculated by the following equation.

$$FBCE_LINE_LENGTH = \lceil pixels_per_line/8 \rceil * FBCE_TARGET_RATE + 2$$

FBCD_INPUT_MODE

- **0** RGB888
- **1** RGB565
- **others** reserved

ENn Enable layer n source data from Frame Buffer Decompression.

LCD +C000h~CFFCh LCD Interface Gamma Correction LUT Registers LCD_GAMMA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GAMMA_LUT2							
Type									R/W							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GAMM_LUT1								GAMMA_LUT0							
Type	R/W								R/W							

GAMMA_LUT0 These Bits Set Gamma LUT 0.
GAMMA_LUT1 These Bits Set Gamma LUT 1.
GAMMA_LUT2 These Bits Set Gamma LUT 2.

LCD +D000h~D3FCh LCD Interface Color Palette LUT Registers LCD_PAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									LUT							
Type									R/W							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LUT															
Type	R/W															

LUT These Bits Set Color Palette in RGB888 format.

LCD +D400h~D4FC LCD Interface Command/Parameter Registers LCD_COMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									C0						COMM[17:16]	
Type									R/W						R/W	
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COMM[15:0]															
Type	R/W															

COMM Command Data and Parameter Data for LCD Module
C0 Write to ROI Command Address if C0 = 1, otherwise write to ROI Data Address

5.21 M3D

5.21.1 General Descriptions

The 3D hardware engine in the MT6516 is designed for OpenGL ES v1.1 Common/Common-Lite Profile Specification and Direct3D Mobile Specification. With this hardware accelerator, can produce the high quality 3D image with high efficiency.

MT6516 3D Hardware Block Diagram :

M3D engine accesses internal memory or external memory through GMC. See Figure 1.1.

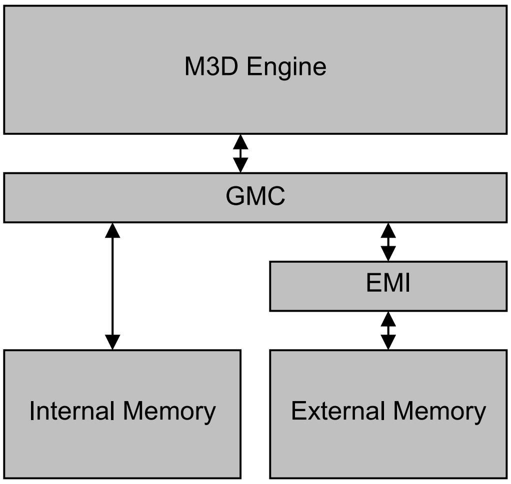


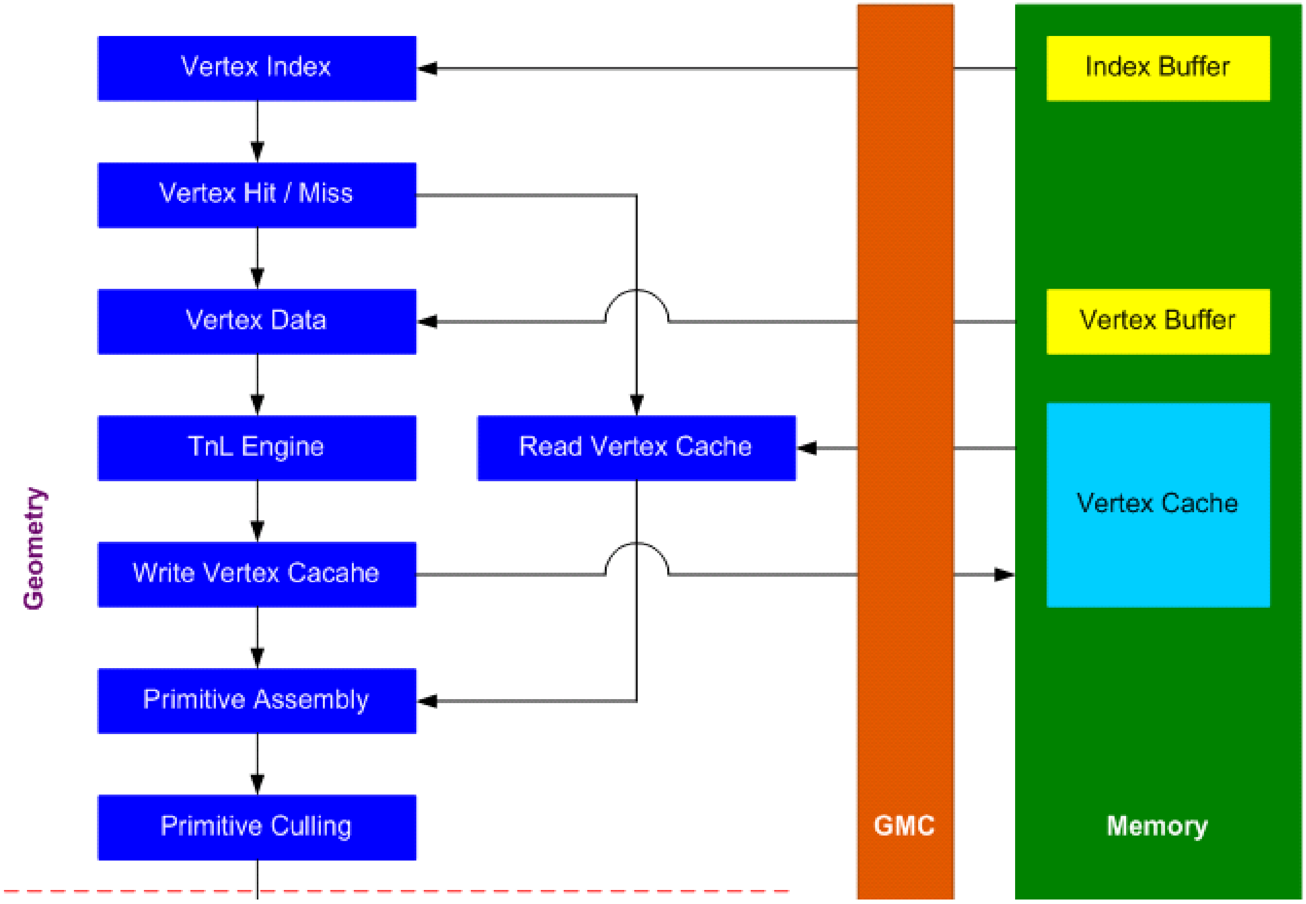
Figure 1.1 MT6516 3D Top architecture

MT6516 3D Key Feature :

- OpenGL ES v1.1 Common/Common-Lite Feature Set with Extensions
- Direct 3D Mobile Feature Set
- Point / Line / Triangle
- Specular Color / Color Sum
- Color Material, Local Viewer and Light Range for Lighting Calculation
- Flat / Gouraud Shading
- Fill Mode (Point / Line / Solid)
- User-Defined Clipping Plane
- Advanced Scan Converter Algorithm
- Texture Coordinate Wrapping
- Texture Addressing Mode : Repeat / Clamp to Edge / Mirror / Clamp / Border
- Perspective Bi- / Tri-linear Texture Filter

- Multi-Texture
- Dot3 Bump Mapping
- Flexible Texture Input Formats
- Advanced Texture Compression
- Support three data formats of frame buffer : RGB565, RGB888, ARGB8888
- Maximum Resolution : 1024 x 1024

The graphics pipeline provides the housepower to efficiently process and render OpenGL ES scenes to a display, taking advantage of available hardware. This figure conceptually illustrates the building blocks of the pipeline :



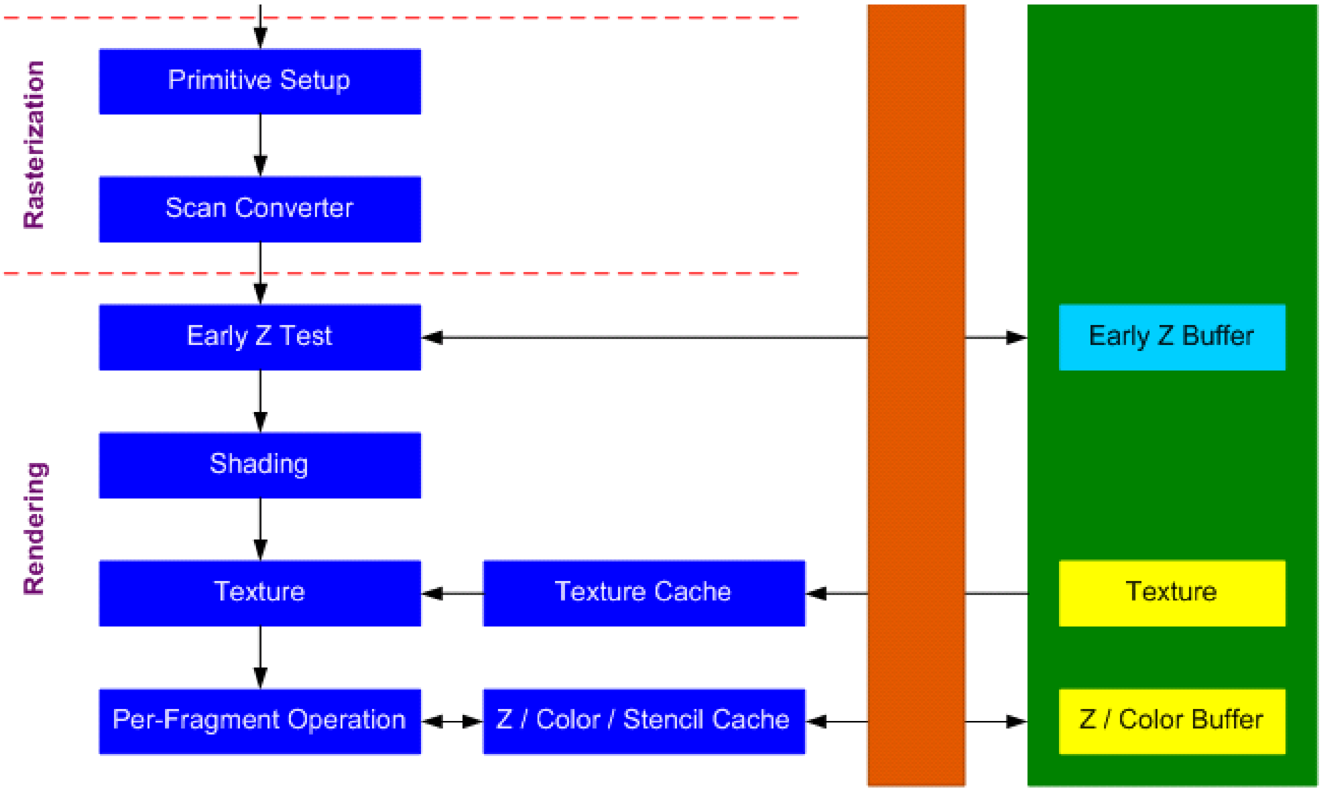


Figure 1.2 MT6516 3D graphics pipeline

5.21.2 Register Definitions

M3D_BASE = 0x

Register Address	Register Function	Acronym
M3D_BASE+000h	M3D trigger register	**M3D_TRIGGER**
M3D_BASE+004h	M3D reset register	**M3D_RESET**
M3D_BASE+008h	M3D status register	**M3D_STATUS**
M3D_BASE+00Ch	M3D interrupt enable register	**M3D_INTEN**
M3D_BASE+010h	M3D interrupt status register	**M3D_INTSTA**
M3D_BASE+014h	M3D fragment cache enable register	**M3D_FRAG_CACHE_ENABLE**
M3D_BASE+018h	M3D fragment cache flush register	**M3D_FRAG_CACHE_FLUSH**
M3D_BASE+01Ch	M3D fragment cache invalidate register	**M3D_FRAG_CACHE_INVAL**
M3D_BASE+020h	M3D fragment cache reset register	**M3D_FRAG_CACHE_RESET**
M3D_BASE+024h	M3D texture cache clear register	**M3D_TEX_CACHE_CLEAR**
M3D_BASE+028h	M3D primitive mode register	**M3D_PRIMITIVE_MODE**
M3D_BASE+02Ch	M3D draw mode register	**M3D_PRIMITIVE_DRAW_MODE**
M3D_BASE+030h	M3D vertex count register	**M3D_PRIMITIVE_COUNT**
M3D_BASE+034h	M3D draw-array first index register	**M3D_DRAW_ARRAY_FIRST**

 © 2009 MediaTek Inc.

M3D_BASE+038h	M3D draw-array end index egister	**M3D_DRAW_ARRAY_END**
M3D_BASE+03Ch	M3D draw-element index type register	**M3D_DRAW_ELEMENT_TYPE**
M3D_BASE+040h	M3D draw-element pointer register	**M3D_DRAW_ELEMENT_POINTER**
M3D_BASE+044h	M3D primitive anti-aliasing register	**M3D_PRIMITIVE_AA**
M3D_BASE+048h	M3D polygon offset enable/disable register	**M3D_POLYGON_OFFSET_ENABLE**
M3D_BASE+04Ch|84Ch	M3D polygon offset type register	**M3D_POLYGON_OFFSET_FACTOR**
M3D_BASE+050h|850h	M3D polygon offset unit register	**M3D_POLYGON_OFFSET_UNITS**
M3D_BASE+054h|854h	M3D line width register	**M3D_LINE_WIDTH**
M3D_BASE+058h	M3D point-sprite register	**M3D_POINT_SPRITE**
M3D_BASE+05Ch	M3D vertex array register	**M3D_VERTEX**
M3D_BASE+060h	M3D vertex array stride register	**M3D_VERTEX_STRIDE_B**
M3D_BASE+064h	M3D vertex array pointer register	**M3D_VERTEX_POINTER**
M3D_BASE+068h	M3D vertex cache pointer register	**M3D_VERTEX_CACHE_POINTER**
M3D_BASE+06Ch	M3D bounding box control register	**M3D_BBOX_EXPAND**
M3D_BASE+070h	M3D normal array type register	**M3D_NORMAL_TYPE**
M3D_BASE+074h	M3D normal array stride register	**M3D_NORMAL_STRIDE_B**
M3D_BASE+078h	M3D normal array pointer register	**M3D_NORMAL_POINTER**
M3D_BASE+07Ch	M3D color array register	**M3D_COLOR_0**
M3D_BASE+080h	M3D color array type/size/input register	**M3D_COLOR_1**
M3D_BASE+084h	M3D color array stride register	**M3D_COLOR_STRIDE_B_0**
M3D_BASE+088h	M3D color array stride register	**M3D_COLOR_STRIDE_B_1**
M3D_BASE+08Ch	M3D color array pointer register	**M3D_COLOR_POINTER_0**
M3D_BASE+090h	M3D color array pointer register	**M3D_COLOR_POINTER_1**
M3D_BASE+094h	M3D debug port 0	**M3D_DBGRD_0**
M3D_BASE+098h	M3D debug port 1	**M3D_DBGRD_1**
M3D_BASE+09Ch	M3D debug port 2	**M3D_DBGRD_2**
M3D_BASE+0A0h	M3D debug port 3	**M3D_DBGRD_3**
M3D_BASE+0A4h	M3D debug port 4	**M3D_DBGRD_4**
M3D_BASE+0A8h	M3D texture coordinate 0 register	**M3D_TEX_COORD_0**
M3D_BASE+0ACh	M3D texture coordinate 1 register	**M3D_TEX_COORD_1**
M3D_BASE+0B0h	M3D texture coordinate 2 register	**M3D_TEX_COORD_2**
M3D_BASE+0B4h	M3D texture array 0 stride register	**M3D_TEX_COORD_STRIDE_B_0**
M3D_BASE+0B8h	M3D texture array 1 stride register	**M3D_TEX_COORD_STRIDE_B_1**
M3D_BASE+0BCh	M3D texture array 2 stride register	**M3D_TEX_COORD_STRIDE_B_2**
M3D_BASE+0C0h	M3D texture array 0 pointer register	**M3D_TEX_COORD_POINTER_0**
M3D_BASE+0C4h	M3D texture array 1 pointer register	**M3D_TEX_COORD_POINTER_1**
M3D_BASE+0C8h	M3D texture array 2 pointer register	**M3D_TEX_COORD_POINTER_2**

© 2009 MediaTek Inc.

M3D_BASE+0CCh	M3D point array register	**M3D_PNT_SIZE_INPUT**
M3D_BASE+0D0h|8D0h	M3D point array size register	**M3D_PNT_SIZE**
M3D_BASE+0D4h	M3D point array stride register	**M3D_PNT_SIZE_STRIDE_B**
M3D_BASE+0D8h	M3D point array pointer register	**M3D_PNT_SIZE_POINTER**
M3D_BASE+0DCh	M3D projection matrix type register	**M3D_PROJ_MATRIX_TYPE**
M3D_BASE+0E0h	M3D user-defined clipping plane enable/disable register	**M3D_CLIP_PLANE_ENABLE**
M3D_BASE+0E4h	M3D normal scale enable/disable register	**M3D_NORMAL_SCALE_ENABLE**
M3D_BASE+0E8h	M3D line last pixel	**M3D_LINE_LAST_PIXEL**
M3D_BASE+0ECh	M3D primitive culling register	**M3D_CULL**
M3D_BASE+0F0h	M3D shading model register	**M3D_SHADE_MODEL**
M3D_BASE+0F4h	M3D/D3DM Lighting Control Register	**M3D_LIGHT_CTRL**
M3D_BASE+0F8h	M3D texture image #0 height/width/type/format register	**M3D_TEX_IMG_0**
M3D_BASE+0FCh	M3D texture image #1 height/width/type/format register	**M3D_TEX_IMG_1**
M3D_BASE+100h	M3D texture image #2 height/width/type/format register	**M3D_TEX_IMG_2**
M3D_BASE+104h	M3D texture control register	**M3D_TEX_CTRL**
M3D_BASE+108h	M3D texture image #0 level #0 address	**M3D_TEX_IMG_PTR_0_0**
M3D_BASE+10Ch	M3D texture image #0 level #1 address	**M3D_TEX_IMG_PTR_0_1**
M3D_BASE+110h	M3D texture image #0 level #2 address	**M3D_TEX_IMG_PTR_0_2**
M3D_BASE+114h	M3D texture image #0 level #3 address	**M3D_TEX_IMG_PTR_0_3**
M3D_BASE+118h	M3D texture image #0 level #4 address	**M3D_TEX_IMG_PTR_0_4**
M3D_BASE+11Ch	M3D texture image #0 level #5 address	**M3D_TEX_IMG_PTR_0_5**
M3D_BASE+120h	M3D texture image #0 level #6 address	**M3D_TEX_IMG_PTR_0_6**
M3D_BASE+124h	M3D texture image #0 level #7 address	**M3D_TEX_IMG_PTR_0_7**
M3D_BASE+128h	M3D texture image #0 level #8 address	**M3D_TEX_IMG_PTR_0_8**
M3D_BASE+12Ch	M3D texture image #1 level #0 address	**M3D_TEX_IMG_PTR_1_0**
M3D_BASE+130h	M3D texture image #1 level #1 address	**M3D_TEX_IMG_PTR_1_1**
M3D_BASE+134h	M3D texture image #1 level #2 address	**M3D_TEX_IMG_PTR_1_2**
M3D_BASE+138h	M3D texture image #1 level #3 address	**M3D_TEX_IMG_PTR_1_3**
M3D_BASE+13Ch	M3D texture image #1 level #4 address	**M3D_TEX_IMG_PTR_1_4**
M3D_BASE+140h	M3D texture image #1 level #5 address	**M3D_TEX_IMG_PTR_1_5**
M3D_BASE+144h	M3D texture image #1 level #6 address	**M3D_TEX_IMG_PTR_1_6**
M3D_BASE+148h	M3D texture image #1 level #7 address	**M3D_TEX_IMG_PTR_1_7**
M3D_BASE+14Ch	M3D texture image #1 level #8 address	**M3D_TEX_IMG_PTR_1_8**
M3D_BASE+150h	M3D texture image #2 level #0 address	**M3D_TEX_IMG_PTR_2_0**
M3D_BASE+154h	M3D texture image #2 level #1 address	**M3D_TEX_IMG_PTR_2_1**

© 2009 MediaTek Inc.

M3D_BASE+158h	M3D texture image #2 level #2 address	**M3D_TEX_IMG_PTR_2_2**
M3D_BASE+15Ch	M3D texture image #2 level #3 address	**M3D_TEX_IMG_PTR_2_3**
M3D_BASE+160h	M3D texture image #2 level #4 address	**M3D_TEX_IMG_PTR_2_4**
M3D_BASE+164h	M3D texture image #2 level #5 address	**M3D_TEX_IMG_PTR_2_5**
M3D_BASE+168h	M3D texture image #2 level #6 address	**M3D_TEX_IMG_PTR_2_6**
M3D_BASE+16Ch	M3D texture image #2 level #7 address	**M3D_TEX_IMG_PTR_2_7**
M3D_BASE+170h	M3D texture image #2 level #8 address	**M3D_TEX_IMG_PTR_2_8**
M3D_BASE+174h|974h	M3D color r register	**M3D_COLOR_R_0**
M3D_BASE+178h|978h	M3D color g register	**M3D_COLOR_G_0**
M3D_BASE+17Ch|97Ch	M3D color b register	**M3D_COLOR_B_0**
M3D_BASE+180h|980h	M3D color a register	**M3D_COLOR_A_0**
M3D_BASE+184h|984h	M3D color r register	**M3D_COLOR_R_1**
M3D_BASE+188h|988h	M3D color g register	**M3D_COLOR_G_1**
M3D_BASE+18Ch|98Ch	M3D color b register	**M3D_COLOR_B_1**
M3D_BASE+190h|990h	M3D color a register	**M3D_COLOR_A_1**
M3D_BASE+194h	M3D drawtex TEX1 cropper region u coord	**M3D_DRAWTEX_CRU_1**
M3D_BASE+198h	M3D scissor test enable/disable register	**M3D_SCISSOR_ENABLE**
M3D_BASE+19Ch	M3D scissor test left boundary register	**M3D_SCISSOR_LEFT**
M3D_BASE+1A0h	M3D scissor test bottom boundary register	**M3D_SCISSOR_BOTTOM**
M3D_BASE+1A4h	M3D scissor test right boundary register	**M3D_SCISSOR_RIGHT**
M3D_BASE+1A8h	M3D scissor test top boundary register	**M3D_SCISSOR_TOP**
M3D_BASE+1ACh	M3D alpha test register	**M3D_ALPHA_TEST**
M3D_BASE+1B0h	M3D stencil test register	**M3D_STENCIL_TEST**
M3D_BASE+1B4h	M3D depth test register	**M3D_DEPTH_TEST**
M3D_BASE+1B8h	M3D blending register	**M3D_BLEND**
M3D_BASE+1BCh	M3D logic operation register	**M3D_LOGIC_OP**
M3D_BASE+1C0h	M3D frame buffer format register	**M3D_FRAME_BUF_FORMAT**
M3D_BASE+1C4h	M3D frame buffer address register	**M3D_FRAME_BUF_ADDR**
M3D_BASE+1C8h	M3D frame buffer width register	**M3D_FRAME_BUF_WIDTH**
M3D_BASE+1CCh	M3D frame buffer height register	**M3D_FRAME_BUF_HEIGHT**
M3D_BASE+1D0h	Reserve for Internal use	**RESERVED**
M3D_BASE+1D4h	M3D depth buffer address	**M3D_DEPTH_BUF_ADDR**
M3D_BASE+1D8h	M3D stencil buffer address	**M3D_STENCIL_BUF_ADDR**
M3D_BASE+1DCh	M3D depth buffer clear value	**M3D_DEPTH_CLEAR_VAL**
M3D_BASE+1E0h	Reserve for Internal use	**RESERVED**
M3D_BASE+1E4h	Reserve for Internal use	**RESERVED**
M3D_BASE+1E8h	M3D earlyz test enable/disable register	**M3D_EARLY_Z_ENABLE**

© 2009 MediaTek Inc.

M3D_BASE+1ECh	M3D earlyz buffer address register	**M3D_EARLY_Z_BUF_ADDR**
M3D_BASE+1F0h	M3D drawtex TEX1 cropper region v coord	**M3D_DRAWTEX_CRV_1**
M3D_BASE+1F4h	M3D drawtex TEX1 cropper region delta u coord	**M3D_DRAWTEX_DCRU_1**
M3D_BASE+1F8h	M3D drawtex TEX1 cropper region delta v coord	**M3D_DRAWTEX_DCRV_1**
M3D_BASE+1FCh	M3D interrupt status write ack	**M3D_INTSTA_WTACK**
M3D_BASE+200h|A00h	M3D texture #0 environment RED color	**M3D_TEX_ENV_COLOR_R_0**
M3D_BASE+204h|A04h	M3D texture #0 environment GREEN color	**M3D_TEX_ENV_COLOR_G_0**
M3D_BASE+208h|A08h	M3D texture #0 environment BLUE color	**M3D_TEX_ENV_COLOR_B_0**
M3D_BASE+20Ch|A0Ch	M3D texture #0 environment ALPHA color	**M3D_TEX_ENV_COLOR_A_0**
M3D_BASE+210h|A10h	M3D texture #1 environment RED color	**M3D_TEX_ENV_COLOR_R_1**
M3D_BASE+214h|A14h	M3D texture #1 environment GREEN color	**M3D_TEX_ENV_COLOR_G_1**
M3D_BASE+218h|A18h	M3D texture #1 environment BLUE color	**M3D_TEX_ENV_COLOR_B_1**
M3D_BASE+21Ch|A1Ch	M3D texture #1 environment ALPHA color	**M3D_TEX_ENV_COLOR_A_1**
M3D_BASE+220h|A20h	M3D texture #2 environment RED color	**M3D_TEX_ENV_COLOR_R_2**
M3D_BASE+224h|A24h	M3D texture #2 environment GREEN color	**M3D_TEX_ENV_COLOR_G_2**
M3D_BASE+228h|A28h	M3D texture #2 environment BLUE color	**M3D_TEX_ENV_COLOR_B_2**
M3D_BASE+22Ch|A2Ch	M3D texture #2 environment ALPHA color	**M3D_TEX_ENV_COLOR_A_2**
M3D_BASE+230h|A30h	M3D texture #0 mipmap LOD bias	**M3D_LOD_BIAS_0**
M3D_BASE+234h|A34h	M3D texture #1 mipmap LOD bias	**M3D_LOD_BIAS_1**
M3D_BASE+238h|A38h	M3D texture #2 mipmap LOD bias	**M3D_LOD_BIAS_2**
M3D_BASE+23Ch	M3D vertex cache hit counter	**M3D_VC_HIT_CNT**
M3D_BASE+240h	M3D texture #0 RGBA scale and texture operation	**M3D_TEX_OP_0**
M3D_BASE+244h	M3D texture #1 RGBA scale and texture operation	**M3D_TEX_OP_1**
M3D_BASE+248h	M3D texture #2 RGBA scale and texture operation	**M3D_TEX_OP_2**
M3D_BASE+24Ch	M3D vertex cache vertex counter	**M3D_VC_VTX_CNT**
M3D_BASE+250h	M3D texture #0 RGBA source/operand	**M3D_TEX_SRC_OPD_0**
M3D_BASE+254h	M3D texture #1 RGBA source/operand	**M3D_TEX_SRC_OPD_1**
M3D_BASE+258h	M3D texture #2 RGBA source/operand	**M3D_TEX_SRC_OPD_2**
M3D_BASE+25Ch	M3D fill mode	**M3D_FILL_MODE**

M3D_BASE+260h|A60h	M3D visible box lower-left x register	**M3D_VB_XLLC**
M3D_BASE+264h|A64h	M3D visible box lower-left y register	**M3D_VB_YLLC**
M3D_BASE+268h|A68h	M3D visible box upper-right x register	**M3D_VB_XURC**
M3D_BASE+26Ch|A6Ch	M3D visible box upper-right y register	**M3D_VB_YURC**
M3D_BASE+270h|A70h	M3D frustum near register	**M3D_FRUSTUM_NEAR**
M3D_BASE+274h|A74h	M3D frustum far register	**M3D_FRUSTUM_FAR**
M3D_BASE+278h|A78h	M3D view port near register	**M3D_VIEWPORT_NEAR**
M3D_BASE+27Ch|A7Ch	M3D view port far register	**M3D_VIEWPORT_FAR**
M3D_BASE+280h|A80h	M3D model view matrix register	**M3D_MODEL_VIEW_M_1**
M3D_BASE+284h|A84h	M3D model view matrix register	**M3D_MODEL_VIEW_M_2**
M3D_BASE+288h|A88h	M3D model view matrix register	**M3D_MODEL_VIEW_M_3**
M3D_BASE+28Ch|A8Ch	M3D model view matrix register	**M3D_MODEL_VIEW_M_4**
M3D_BASE+290h|A90h	M3D model view matrix register	**M3D_MODEL_VIEW_M_5**
M3D_BASE+294h|A94h	M3D model view matrix register	**M3D_MODEL_VIEW_M_6**
M3D_BASE+298h|A98h	M3D model view matrix register	**M3D_MODEL_VIEW_M_7**
M3D_BASE+29Ch|A9Ch	M3D model view matrix register	**M3D_MODEL_VIEW_M_8**
M3D_BASE+2A0h|AA0h	M3D model view matrix register	**M3D_MODEL_VIEW_M_9**
M3D_BASE+2A4h|AA4h	M3D model view matrix register	**M3D_MODEL_VIEW_M_10**
M3D_BASE+2A8h|AA8h	M3D model view matrix register	**M3D_MODEL_VIEW_M_11**
M3D_BASE+2ACh|AAch	M3D model view matrix register	**M3D_MODEL_VIEW_M_12**
M3D_BASE+2B0h|AB0h	M3D model view matrix register	**M3D_MODEL_VIEW_M_13**
M3D_BASE+2B4h|AB4h	M3D model view matrix register	**M3D_MODEL_VIEW_M_14**
M3D_BASE+2B8h|AB8h	M3D model view matrix register	**M3D_MODEL_VIEW_M_15**
M3D_BASE+2BCh|ABCh	M3D model view matrix register	**M3D_MODEL_VIEW_M_16**
M3D_BASE+2C0h|AC0h	M3D vp matrix register	**M3D_PV_1**
M3D_BASE+2C4h|AC4h	M3D vp matrix register	**M3D_PV_2**
M3D_BASE+2C8h|AC8h	M3D vp matrix register	**M3D_PV_3**
M3D_BASE+2CCh|ACCh	M3D vp matrix register	**M3D_PV_4**
M3D_BASE+2D0h|AD0h	M3D vp matrix register	**M3D_PV_5**
M3D_BASE+2D4h|AD4h	M3D vp matrix register	**M3D_PV_6**
M3D_BASE+2D8h|AD8h	M3D vp matrix register	**M3D_PV_7**
M3D_BASE+2DCh|ADCh	M3D vp matrix register	**M3D_PV_8**
M3D_BASE+2E0h|AE0h	M3D vp matrix register	**M3D_PV_9**
M3D_BASE+2E4h|AE4h	M3D vp matrix register	**M3D_PV_10**
M3D_BASE+2E8h|AE8h	M3D vp matrix register	**M3D_PV_11**
M3D_BASE+2ECh|AECh	M3D vp matrix register	**M3D_PV_12**
M3D_BASE+2F0h|AF0h	M3D vp matrix register	**M3D_PV_13**
M3D_BASE+2F4h|AF4h	M3D vp matrix register	**M3D_PV_14**

M3D_BASE+2F8h|AF8h	M3D vp matrix register	**M3D_PV_15**
M3D_BASE+2FCh|AFCh	M3D vp matrix register	**M3D_PV_16**
M3D_BASE+300h|B00h	M3D user-defined clipping plane matrix register	**M3D_C_N_X**
M3D_BASE+304h|B04h	M3D user-defined clipping plane matrix register	**M3D_C_N_Y**
M3D_BASE+308h|B08h	M3D user-defined clipping plane matrix register	**M3D_C_N_Z**
M3D_BASE+30Ch|B0Ch	M3D user-defined clipping plane matrix register	**M3D_C_N_W**
M3D_BASE+310h|B10h	M3D texture #0 border color RED	**M3D_TEX_BORDER_R_0**
M3D_BASE+314h|B14h	M3D texture #0 border color GREEN	**M3D_TEX_BORDER_G_0**
M3D_BASE+318h|B18h	M3D texture #0 border color BLUE	**M3D_TEX_BORDER_B_0**
M3D_BASE+31Ch|B1Ch	M3D texture #0 border color ALPHA	**M3D_TEX_BORDER_A_0**
M3D_BASE+320h|B20h	M3D texture #1 border color RED	**M3D_TEX_BORDER_R_1**
M3D_BASE+324h|B24h	M3D texture #1 border color GREEN	**M3D_TEX_BORDER_G_1**
M3D_BASE+328h|B28h	M3D texture #1 border color BLUE	**M3D_TEX_BORDER_B_1**
M3D_BASE+32Ch|B2Ch	M3D texture #1 border color ALPHA	**M3D_TEX_BORDER_A_1**
M3D_BASE+330h|B30h	M3D texture #2 border color RED	**M3D_TEX_BORDER_R_2**
M3D_BASE+334h|B34h	M3D texture #2 border color GREEN	**M3D_TEX_BORDER_G_2**
M3D_BASE+338h|B38h	M3D texture #2 border color BLUE	**M3D_TEX_BORDER_B_2**
M3D_BASE+33Ch|B3Ch	M3D texture #2 border color ALPHA	**M3D_TEX_BORDER_A_2**
M3D_BASE+340h|B40h	M3D texture coordinate matrix register	**M3D_TEX_M_0_1**
M3D_BASE+344h|B44h	M3D texture coordinate matrix register	**M3D_TEX_M_0_2**
M3D_BASE+348h|B48h	M3D texture coordinate matrix register	**M3D_TEX_M_0_3**
M3D_BASE+34Ch|B4Ch	M3D texture coordinate matrix register	**M3D_TEX_M_0_4**
M3D_BASE+350h|B50h	M3D texture coordinate matrix register	**M3D_TEX_M_0_5**
M3D_BASE+354h|B54h	M3D texture coordinate matrix register	**M3D_TEX_M_0_6**
M3D_BASE+358h|B58h	M3D texture coordinate matrix register	**M3D_TEX_M_0_7**
M3D_BASE+35Ch|B5Ch	M3D texture coordinate matrix register	**M3D_TEX_M_0_8**
M3D_BASE+360h|B60h	M3D texture coordinate matrix register	**M3D_TEX_M_0_9**
M3D_BASE+364h|B64h	M3D texture coordinate matrix register	**M3D_TEX_M_0_10**
M3D_BASE+368h|B68h	M3D texture coordinate matrix register	**M3D_TEX_M_0_11**
M3D_BASE+36Ch|B6Ch	M3D texture coordinate matrix register	**M3D_TEX_M_0_12**
M3D_BASE+370h|B70h	M3D texture coordinate matrix register	**M3D_TEX_M_1_1**
M3D_BASE+374h|B74h	M3D texture coordinate matrix register	**M3D_TEX_M_1_2**
M3D_BASE+378h|B78h	M3D texture coordinate matrix register	**M3D_TEX_M_1_3**
M3D_BASE+37Ch|B7Ch	M3D texture coordinate matrix register	**M3D_TEX_M_1_4**
M3D_BASE+380h|B80h	M3D texture coordinate matrix register	**M3D_TEX_M_1_5**

© 2009 MediaTek Inc.

M3D_BASE+384h|B84h	M3D texture coordinate matrix register	**M3D_TEX_M_1_6**
M3D_BASE+388h|B88h	M3D texture coordinate matrix register	**M3D_TEX_M_1_7**
M3D_BASE+38Ch|B8Ch	M3D texture coordinate matrix register	**M3D_TEX_M_1_8**
M3D_BASE+390h|B90h	M3D texture coordinate matrix register	**M3D_TEX_M_1_9**
M3D_BASE+394h|B94h	M3D texture coordinate matrix register	**M3D_TEX_M_1_10**
M3D_BASE+398h|B98h	M3D texture coordinate matrix register	**M3D_TEX_M_1_11**
M3D_BASE+39Ch|B9Ch	M3D texture coordinate matrix register	**M3D_TEX_M_1_12**
M3D_BASE+3A0h|BA0h	M3D texture coordinate matrix register	**M3D_TEX_M_2_1**
M3D_BASE+3A4h|BA4h	M3D texture coordinate matrix register	**M3D_TEX_M_2_2**
M3D_BASE+3A8h|BA8h	M3D texture coordinate matrix register	**M3D_TEX_M_2_3**
M3D_BASE+3ACh|BACh	M3D texture coordinate matrix register	**M3D_TEX_M_2_4**
M3D_BASE+3B0h|BB0h	M3D texture coordinate matrix register	**M3D_TEX_M_2_5**
M3D_BASE+3B4h|BB4h	M3D texture coordinate matrix register	**M3D_TEX_M_2_6**
M3D_BASE+3B8h|BB8h	M3D texture coordinate matrix register	**M3D_TEX_M_2_7**
M3D_BASE+3BCh|BBCh	M3D texture coordinate matrix register	**M3D_TEX_M_2_8**
M3D_BASE+3C0h|BC0h	M3D texture coordinate matrix register	**M3D_TEX_M_2_9**
M3D_BASE+3C4h|BC4h	M3D texture coordinate matrix register	**M3D_TEX_M_2_10**
M3D_BASE+3C8h|BC8h	M3D texture coordinate matrix register	**M3D_TEX_M_2_11**
M3D_BASE+3CCh|BCCh	M3D texture coordinate matrix register	**M3D_TEX_M_2_12**
M3D_BASE+3D0h|BD0h	D3DM Light Range of Light Source 0 Register	**M3D_LIGHT_RANGE_0**
M3D_BASE+3D4h|BD4h	D3DM Light Range of Light Source 1 Register	**M3D_LIGHT_RANGE_1**
M3D_BASE+3D8h|BD8h	D3DM Light Range of Light Source 2 Register	**M3D_LIGHT_RANGE_2**
M3D_BASE+3DCh|BDCh	D3DM Light Range of Light Source 3 Register	**M3D_LIGHT_RANGE_3**
M3D_BASE+3E0h|BE0h	D3DM Light Range of Light Source 4 Register	**M3D_LIGHT_RANGE_4**
M3D_BASE+3E4h|BE4h	D3DM Light Range of Light Source 5 Register	**M3D_LIGHT_RANGE_5**
M3D_BASE+3E8h|BE8h	D3DM Light Range of Light Source 6 Register	**M3D_LIGHT_RANGE_6**
M3D_BASE+3ECh|BECh	D3DM Light Range of Light Source 7 Register	**M3D_LIGHT_RANGE_7**
M3D_BASE+3F0h	M3D drawtex TEX2 cropper region u coord	**M3D_DRAWTEX_CRU_2**
M3D_BASE+3F4h	M3D drawtex TEX2 cropper region v coord	**M3D_DRAWTEX_CRV_2**
M3D_BASE+3F8h	M3D drawtex TEX2 cropper region delta u coord	**M3D_DRAWTEX_DCRU_2**

© 2009 MediaTek Inc.

M3D_BASE+3FCh	M3D drawtex TEX2 cropper region delta v coord	**M3D_DRAWTEX_DCRV_2**
M3D_BASE+400h|C00h	M3D normal matrix register	**M3D_NORMAL_N_1**
M3D_BASE+404h|C04h	M3D normal matrix register	**M3D_NORMAL_N_2**
M3D_BASE+408h|C08h	M3D normal matrix register	**M3D_NORMAL_N_3**
M3D_BASE+40Ch|C0Ch	M3D normal matrix register	**M3D_NORMAL_N_4**
M3D_BASE+410h|C10h	M3D normal matrix register	**M3D_NORMAL_N_5**
M3D_BASE+414h|C14h	M3D normal matrix register	**M3D_NORMAL_N_6**
M3D_BASE+418h|C18h	M3D normal matrix register	**M3D_NORMAL_N_7**
M3D_BASE+41Ch|C1Ch	M3D normal matrix register	**M3D_NORMAL_N_8**
M3D_BASE+420h|C20h	M3D normal matrix register	**M3D_NORMAL_N_9**
M3D_BASE+424h|C24h	M3D normal scale register	**M3D_NORMAL_SCALE**
M3D_BASE+428h|C28h	M3D fog color's red component register	**M3D_FOG_COLOR_R**
M3D_BASE+42Ch|C2Ch	M3D fog color's green component register	**M3D_FOG_COLOR_G**
M3D_BASE+430h|C30h	M3D fog color's blue component register	**M3D_FOG_COLOR_B**
M3D_BASE+434h|C34h	M3D fog color's alpha component register	**M3D_FOG_COLOR_A**
M3D_BASE+438h|C38h	Reserve for Internal use	**RESERVED**
M3D_BASE+43Ch	Pixel counter	**M3D_PXL_CNT**
M3D_BASE+440h|C40h	Red Component of Ambient Color of Material Register	**M3D_A_CM_R**
M3D_BASE+444h|C44h	Green Component of Ambient Color of Material Register	**M3D_A_CM_G**
M3D_BASE+448h|C48h	Blue Component of Ambient Color of Material Register	**M3D_A_CM_B**
M3D_BASE+44Ch	Texture cache hit counter	**M3D_TXCACHE_CNT**
M3D_BASE+450h|C50h	Red Component of Diffusion Color of Material Register	**M3D_D_CM_R**
M3D_BASE+454h|C54h	Green Component of Diffusion Color of Material Register	**M3D_D_CM_G**
M3D_BASE+458h|C58h	Blue Component of Diffusion Color of Material Register	**M3D_D_CM_B**
M3D_BASE+45Ch|C5Ch	Alpha Component of Diffusion Color of Material Register	**M3D_D_CM_A**
M3D_BASE+460h|C60h	Red Component of Specular Color of Material Register	**M3D_s_CM_R**
M3D_BASE+464h|C64h	Green Component of Specular Color of Material Register	**M3D_S_CM_G**
M3D_BASE+468h|C68h	Blue Component of Specular Color of Material Register	**M3D_S_CM_B**
M3D_BASE+46Ch|C6Ch	Alpha Component of Specular Color of	**M3D_S_CM_A**

	Material Register	
M3D_BASE+470h|C70h	Red Component of Emission Color of Material Register	**M3D_E_CM_R**
M3D_BASE+474h|C74h	Green Component of Emission Color of Material Register	**M3D_E_CM_G**
M3D_BASE+478h|C78h	Blue Component of Emission Color of Material Register	**M3D_E_CM_B**
M3D_BASE+47Ch	Z cache hit counter	**M3D_ZC_CNT**
M3D_BASE+480h|C80h	Red Component of Ambient Intensity of Light Source 0 Register	**M3D_A_CL_0_R**
M3D_BASE+484h|C84h	Green Component of Ambient Intensity of Light Source 0 Register	**M3D_A_CL_0_G**
M3D_BASE+488h|C88h	Blue Component of Ambient Intensity of Light Source 0 Register	**M3D_A_CL_0_B**
M3D_BASE+48Ch|C8Ch	Alpha Component of Ambient Intensity of Light Source 0 Register	**M3D_A_CL_0_A**
M3D_BASE+490h|C90h	Red Component of Ambient Intensity of Light Source 1 Register	**M3D_A_CL_1_R**
M3D_BASE+494h|C94h	Green Component of Ambient Intensity of Light Source 1 Register	**M3D_A_CL_1_G**
M3D_BASE+498h|C98h	Blue Component of Ambient Intensity of Light Source 1 Register	**M3D_A_CL_1_B**
M3D_BASE+49Ch|C9Ch	Alpha Component of Ambient Intensity of Light Source 1 Register	**M3D_A_CL_1_A**
M3D_BASE+4A0h|CA0h	Red Component of Ambient Intensity of Light Source 2 Register	**M3D_A_CL_2_R**
M3D_BASE+4A4h|CA4h	Green Component of Ambient Intensity of Light Source 2 Register	**M3D_A_CL_2_G**
M3D_BASE+4A8h|CA8h	Blue Component of Ambient Intensity of Light Source 2 Register	**M3D_A_CL_2_B**
M3D_BASE+4ACh|CACh	Alpha Component of Ambient Intensity of Light Source 2 Register	**M3D_A_CL_2_A**
M3D_BASE+4B0h|CB0h	Red Component of Ambient Intensity of Light Source 3 Register	**M3D_A_CL_3_R**
M3D_BASE+4B4h|CB4h	Green Component of Ambient Intensity of Light Source 3 Register	**M3D_A_CL_3_G**
M3D_BASE+4B8h|CB8h	Blue Component of Ambient Intensity of Light Source 3 Register	**M3D_A_CL_3_B**
M3D_BASE+4BCh|CBCh	Alpha Component of Ambient Intensity of Light Source 3 Register	**M3D_A_CL_3_A**
M3D_BASE+4C0h|CC0h	Red Component of Ambient Intensity of Light Source 4 Register	**M3D_A_CL_4_R**
M3D_BASE+4C4h|CC4h	Green Component of Ambient Intensity of Light Source 4 Register	**M3D_A_CL_4_G**

M3D_BASE+4C8h|CC8h	Blue Component of Ambient Intensity of Light Source 4 Register	**M3D_A_CL_4_B**
M3D_BASE+4CCh|CCCh	Alpha Component of Ambient Intensity of Light Source 4 Register	**M3D_A_CL_4_A**
M3D_BASE+4D0h|CD0h	Red Component of Ambient Intensity of Light Source 5 Register	**M3D_A_CL_5_R**
M3D_BASE+4D4h|CD4h	Green Component of Ambient Intensity of Light Source 5 Register	**M3D_A_CL_5_G**
M3D_BASE+4D8h|CD8h	Blue Component of Ambient Intensity of Light Source 5 Register	**M3D_A_CL_5_B**
M3D_BASE+4DCh|CDCh	Alpha Component of Ambient Intensity of Light Source 5 Register	**M3D_A_CL_5_A**
M3D_BASE+4E0h|CE0h	Red Component of Ambient Intensity of Light Source 6 Register	**M3D_A_CL_6_R**
M3D_BASE+4E4h|CE4h	Green Component of Ambient Intensity of Light Source 6 Register	**M3D_A_CL_6_G**
M3D_BASE+4E8h|CE8h	Blue Component of Ambient Intensity of Light Source 6 Register	**M3D_A_CL_6_B**
M3D_BASE+4ECh|CECh	Alpha Component of Ambient Intensity of Light Source 6 Register	**M3D_A_CL_6_A**
M3D_BASE+4F0h|CF0h	Red Component of Ambient Intensity of Light Source 7 Register	**M3D_A_CL_7_R**
M3D_BASE+4F4h|CF4h	Green Component of Ambient Intensity of Light Source 7 Register	**M3D_A_CL_7_G**
M3D_BASE+4F8h|CF8h	Blue Component of Ambient Intensity of Light Source 7 Register	**M3D_A_CL_7_B**
M3D_BASE+4FCh|CFCh	Alpha Component of Ambient Intensity of Light Source 7 Register	**M3D_A_CL_7_A**
M3D_BASE+500h|D00h	Red Component of Diffusion Intensity of Light Source 0 Register	**M3D_D_CL_0_R**
M3D_BASE+504h|D04h	Green Component of Diffusion Intensity of Light Source 0 Register	**M3D_D_CL_0_G**
M3D_BASE+508h|D08h	Blue Component of Diffusion Intensity of Light Source 0 Register	**M3D_D_CL_0_B**
M3D_BASE+50Ch|D0Ch	Alpha Component of Diffusion Intensity of Light Source 0 Register	**M3D_D_CL_0_A**
M3D_BASE+510h|D10h	Red Component of Diffusion Intensity of Light Source 1 Register	**M3D_D_CL_1_R**
M3D_BASE+514h|D14h	Green Component of Diffusion Intensity of Light Source 1 Register	**M3D_D_CL_1_G**
M3D_BASE+518h|D18h	Blue Component of Diffusion Intensity of Light Source 1 Register	**M3D_D_CL_1_B**
M3D_BASE+51Ch|D1Ch	Alpha Component of Diffusion Intensity of Light Source 1 Register	**M3D_D_CL_1_A**

© 2009 MediaTek Inc.

M3D_BASE+520h|D20h	Red Component of Diffusion Intensity of Light Source 2 Register	**M3D_D_CL_2_R**
M3D_BASE+524h|D24h	Green Component of Diffusion Intensity of Light Source 2 Register	**M3D_D_CL_2_G**
M3D_BASE+528h|D28h	Blue Component of Diffusion Intensity of Light Source 2 Register	**M3D_D_CL_2_B**
M3D_BASE+52Ch|D2Ch	Alpha Component of Diffusion Intensity of Light Source 2 Register	**M3D_D_CL_2_A**
M3D_BASE+530h|D30h	Red Component of Diffusion Intensity of Light Source 3 Register	**M3D_D_CL_3_R**
M3D_BASE+534h|D34h	Green Component of Diffusion Intensity of Light Source 3 Register	**M3D_D_CL_3_G**
M3D_BASE+538h|D38h	Blue Component of Diffusion Intensity of Light Source 3 Register	**M3D_D_CL_3_B**
M3D_BASE+53Ch|D3Ch	Alpha Component of Diffusion Intensity of Light Source 3 Register	**M3D_D_CL_3_A**
M3D_BASE+540h|D40h	Red Component of Diffusion Intensity of Light Source 4 Register	**M3D_D_CL_4_R**
M3D_BASE+544h|D44h	Green Component of Diffusion Intensity of Light Source 4 Register	**M3D_D_CL_4_G**
M3D_BASE+548h|D48h	Blue Component of Diffusion Intensity of Light Source 4 Register	**M3D_D_CL_4_B**
M3D_BASE+54Ch|D4Ch	Alpha Component of Diffusion Intensity of Light Source 4 Register	**M3D_D_CL_4_A**
M3D_BASE+550h|D50h	Red Component of Diffusion Intensity of Light Source 5 Register	**M3D_D_CL_5_R**
M3D_BASE+554h|D54h	Green Component of Diffusion Intensity of Light Source 5 Register	**M3D_D_CL_5_G**
M3D_BASE+558h|D58h	Blue Component of Diffusion Intensity of Light Source 5 Register	**M3D_D_CL_5_B**
M3D_BASE+55Ch|D5Ch	Alpha Component of Diffusion Intensity of Light Source 5 Register	**M3D_D_CL_5_A**
M3D_BASE+560h|D60h	Red Component of Diffusion Intensity of Light Source 6 Register	**M3D_D_CL_6_R**
M3D_BASE+564h|D64h	Green Component of Diffusion Intensity of Light Source 6 Register	**M3D_D_CL_6_G**
M3D_BASE+568h|D68h	Blue Component of Diffusion Intensity of Light Source 6 Register	**M3D_D_CL_6_B**
M3D_BASE+56Ch|D6Ch	Alpha Component of Diffusion Intensity of Light Source 6 Register	**M3D_D_CL_6_A**
M3D_BASE+570h|D70h	Red Component of Diffusion Intensity of Light Source 7 Register	**M3D_D_CL_7_R**
M3D_BASE+574h|D74h	Green Component of Diffusion Intensity of Light Source 7 Register	**M3D_D_CL_7_G**

M3D_BASE+578h|D78h	Blue Component of Diffusion Intensity of Light Source 7 Register	**M3D_D_CL_7_B**
M3D_BASE+57Ch|D7Ch	Alpha Component of Diffusion Intensity of Light Source 7 Register	**M3D_D_CL_7_A**
M3D_BASE+580h|D80h	Red Component of Specular Intensity of Light Source 0 Register	**M3D_S_CL_0_R**
M3D_BASE+584h|D84h	Green Component of Specular Intensity of Light Source 0 Register	**M3D_S_CL_0_G**
M3D_BASE+588h|D88h	Blue Component of Specular Intensity of Light Source 0 Register	**M3D_S_CL_0_B**
M3D_BASE+58Ch|D8Ch	Alpha Component of Specular Intensity of Light Source 0 Register	**M3D_S_CL_0_A**
M3D_BASE+590h|D90h	Red Component of Specular Intensity of Light Source 1 Register	**M3D_S_CL_1_R**
M3D_BASE+594h|D94h	Green Component of Specular Intensity of Light Source 1 Register	**M3D_S_CL_1_G**
M3D_BASE+598h|D98h	Blue Component of Specular Intensity of Light Source 1 Register	**M3D_S_CL_1_B**
M3D_BASE+59Ch|D9Ch	Alpha Component of Specular Intensity of Light Source 1 Register	**M3D_S_CL_1_A**
M3D_BASE+5A0h|DA0h	Red Component of Specular Intensity of Light Source 2 Register	**M3D_S_CL_2_R**
M3D_BASE+5A4h|DA4h	Green Component of Specular Intensity of Light Source 2 Register	**M3D_S_CL_2_G**
M3D_BASE+5A8h|DA8h	Blue Component of Specular Intensity of Light Source 2 Register	**M3D_S_CL_2_B**
M3D_BASE+5ACh|DACh	Alpha Component of Specular Intensity of Light Source 2 Register	**M3D_S_CL_2_A**
M3D_BASE+5B0h|DB0h	Red Component of Specular Intensity of Light Source 3 Register	**M3D_S_CL_3_R**
M3D_BASE+5B4h|DB4h	Green Component of Specular Intensity of Light Source 3 Register	**M3D_S_CL_3_G**
M3D_BASE+5B8h|DB8h	Blue Component of Specular Intensity of Light Source 3 Register	**M3D_S_CL_3_B**
M3D_BASE+5BCh|DBCh	Alpha Component of Specular Intensity of Light Source 3 Register	**M3D_S_CL_3_A**
M3D_BASE+5C0h|DC0h	Red Component of Specular Intensity of Light Source 4 Register	**M3D_S_CL_4_R**
M3D_BASE+5C4h|DC4h	Green Component of Specular Intensity of Light Source 4 Register	**M3D_S_CL_4_G**
M3D_BASE+5C8h|DC8h	Blue Component of Specular Intensity of Light Source 4 Register	**M3D_S_CL_4_B**
M3D_BASE+5CCh|DCCh	Alpha Component of Specular Intensity of Light Source 4 Register	**M3D_S_CL_4_A**

© 2009 MediaTek Inc.

M3D_BASE+5D0h|DD0h	Red Component of Specular Intensity of Light Source 5 Register	**M3D_S_CL_5_R**
M3D_BASE+5D4h|DD4h	Green Component of Specular Intensity of Light Source 5 Register	**M3D_S_CL_5_G**
M3D_BASE+5D8h|DD8h	Blue Component of Specular Intensity of Light Source 5 Register	**M3D_S_CL_5_B**
M3D_BASE+5DCh|DDCh	Alpha Component of Specular Intensity of Light Source 5 Register	**M3D_S_CL_5_A**
M3D_BASE+5E0h|DE0h	Red Component of Specular Intensity of Light Source 6 Register	**M3D_S_CL_6_R**
M3D_BASE+5E4h|DE4h	Green Component of Specular Intensity of Light Source 6 Register	**M3D_S_CL_6_G**
M3D_BASE+5E8h|DE8h	Blue Component of Specular Intensity of Light Source 6 Register	**M3D_S_CL_6_B**
M3D_BASE+5ECh|DECh	Alpha Component of Specular Intensity of Light Source 6 Register	**M3D_S_CL_6_A**
M3D_BASE+5F0h|DF0h	Red Component of Specular Intensity of Light Source 7 Register	**M3D_S_CL_7_R**
M3D_BASE+5F4h|DF4h	Green Component of Specular Intensity of Light Source 7 Register	**M3D_S_CL_7_G**
M3D_BASE+5F8h|DF8h	Blue Component of Specular Intensity of Light Source 7 Register	**M3D_S_CL_7_B**
M3D_BASE+5FCh|DFCh	Alpha Component of Specular Intensity of Light Source 7 Register	**M3D_S_CL_7_A**
M3D_BASE+600h|E00h	X Coordinate of Position of Light Source 0 Register	**M3D_P_PL_0_X**
M3D_BASE+604h|E04h	Y Coordinate of Position of Light Source 0 Register	**M3D_P_PL_0_Y**
M3D_BASE+608h|E08h	Z Coordinate of Position of Light Source 0 Register	**M3D_P_PL_0_Z**
M3D_BASE+60Ch|E0Ch	W Coordinate of Position of Light Source 0 Register	**M3D_P_PL_0_W**
M3D_BASE+610h|E10h	X Coordinate of Position of Light Source 1 Register	**M3D_P_PL_1_X**
M3D_BASE+614h|E14h	Y Coordinate of Position of Light Source 1 Register	**M3D_P_PL_1_Y**
M3D_BASE+618h|E18h	Z Coordinate of Position of Light Source 1 Register	**M3D_P_PL_1_Z**
M3D_BASE+61Ch|E1Ch	W Coordinate of Position of Light Source 1 Register	**M3D_P_PL_1_W**
M3D_BASE+620h|E20h	X Coordinate of Position of Light Source 2 Register	**M3D_P_PL_2_X**
M3D_BASE+624h|E24h	Y Coordinate of Position of Light Source 2 Register	**M3D_P_PL_2_Y**

 © 2009 MediaTek Inc.

M3D_BASE+628h|E28h	Z Coordinate of Position of Light Source 2 Register	**M3D_P_PL_2_Z**
M3D_BASE+62Ch|E2Ch	W Coordinate of Position of Light Source 2 Register	**M3D_P_PL_2_W**
M3D_BASE+630h|E30h	X Coordinate of Position of Light Source 3 Register	**M3D_P_PL_3_X**
M3D_BASE+634h|E34h	Y Coordinate of Position of Light Source 3 Register	**M3D_P_PL_3_Y**
M3D_BASE+638h|E38h	Z Coordinate of Position of Light Source 3 Register	**M3D_P_PL_3_Z**
M3D_BASE+63Ch|E3Ch	W Coordinate of Position of Light Source 3 Register	**M3D_P_PL_3_W**
M3D_BASE+640h|E40h	X Coordinate of Position of Light Source 4 Register	**M3D_P_PL_4_X**
M3D_BASE+644h|E44h	Y Coordinate of Position of Light Source 4 Register	**M3D_P_PL_4_Y**
M3D_BASE+648h|E48h	Z Coordinate of Position of Light Source 4 Register	**M3D_P_PL_4_Z**
M3D_BASE+64Ch|E4Ch	W Coordinate of Position of Light Source 4 Register	**M3D_P_PL_4_W**
M3D_BASE+650h|E50h	X Coordinate of Position of Light Source 5 Register	**M3D_P_PL_5_X**
M3D_BASE+654h|E54h	Y Coordinate of Position of Light Source 5 Register	**M3D_P_PL_5_Y**
M3D_BASE+658h|E58h	Z Coordinate of Position of Light Source 5 Register	**M3D_P_PL_5_Z**
M3D_BASE+65Ch|E5Ch	W Coordinate of Position of Light Source 5 Register	**M3D_P_PL_5_W**
M3D_BASE+660h|E60h	X Coordinate of Position of Light Source 6 Register	**M3D_P_PL_6_X**
M3D_BASE+664h|E64h	Y Coordinate of Position of Light Source 6 Register	**M3D_P_PL_6_Y**
M3D_BASE+668h|E68h	Z Coordinate of Position of Light Source 6 Register	**M3D_P_PL_6_Z**
M3D_BASE+66Ch|E6Ch	W Coordinate of Position of Light Source 6 Register	**M3D_P_PL_6_W**
M3D_BASE+670h|E70h	X Coordinate of Position of Light Source 7 Register	**M3D_P_PL_7_X**
M3D_BASE+674h|E74h	Y Coordinate of Position of Light Source 7 Register	**M3D_P_PL_7_Y**
M3D_BASE+678h|E78h	Z Coordinate of Position of Light Source 7 Register	**M3D_P_PL_7_Z**
M3D_BASE+67Ch|E7Ch	W Coordinate of Position of Light Source 7 Register	**M3D_P_PL_7_W**

© 2009 MediaTek Inc.

M3D_BASE+680h|E80h	X Direction of Spotlight for Light Source 0 Register	**M3D_S_DL_0_X**
M3D_BASE+684h|E84h	Y Direction of Spotlight for Light Source 0 Register	**M3D_S_DL_0_Y**
M3D_BASE+688h|E88h	Z Direction of Spotlight for Light Source 0 Register	**M3D_S_DL_0_Z**
M3D_BASE+68Ch|E8Ch	Spotlight Exponent for Light Source 0 Register	**M3D_S_RL_0**
M3D_BASE+690h|E90h	X Direction of Spotlight for Light Source 1 Register	**M3D_S_DL_1_X**
M3D_BASE+694h|E94h	Y Direction of Spotlight for Light Source 1 Register	**M3D_S_DL_1_Y**
M3D_BASE+698h|E98h	Z Direction of Spotlight for Light Source 1 Register	**M3D_S_DL_1_Z**
M3D_BASE+69Ch|E9Ch	Spotlight Exponent for Light Source 1 Register	**M3D_S_RL_1**
M3D_BASE+6A0h|EA0h	X Direction of Spotlight for Light Source 2 Register	**M3D_S_DL_2_X**
M3D_BASE+6A4h|EA4h	Y Direction of Spotlight for Light Source 2 Register	**M3D_S_DL_2_Y**
M3D_BASE+6A8h|EA8h	Z Direction of Spotlight for Light Source 2 Register	**M3D_S_DL_2_Z**
M3D_BASE+6ACh|EACh	Spotlight Exponent for Light Source 2 Register	**M3D_S_RL_2**
M3D_BASE+6B0h|EB0h	X Direction of Spotlight for Light Source 3 Register	**M3D_S_DL_3_X**
M3D_BASE+6B4h|EB4h	Y Direction of Spotlight for Light Source 3 Register	**M3D_S_DL_3_Y**
M3D_BASE+6B8h|EB8h	Z Direction of Spotlight for Light Source 3 Register	**M3D_S_DL_3_Z**
M3D_BASE+6BCh|EBCh	Spotlight Exponent for Light Source 3 Register	**M3D_S_RL_3**
M3D_BASE+6C0h|EC0h	X Direction of Spotlight for Light Source 4 Register	**M3D_S_DL_4_X**
M3D_BASE+6C4h|EC4h	Y Direction of Spotlight for Light Source 4 Register	**M3D_S_DL_4_Y**
M3D_BASE+6C8h|EC8h	Z Direction of Spotlight for Light Source 4 Register	**M3D_S_DL_4_Z**
M3D_BASE+6CCh|ECCh	Spotlight Exponent for Light Source 4 Register	**M3D_S_RL_4**
M3D_BASE+6D0h|ED0h	X Direction of Spotlight for Light Source 5 Register	**M3D_S_DL_5_X**
M3D_BASE+6D4h|ED4h	Y Direction of Spotlight for Light Source 5 Register	**M3D_S_DL_5_Y**

© 2009 MediaTek Inc.

M3D_BASE+6D8h|ED8h	Z Direction of Spotlight for Light Source 5 Register	**M3D_S_DL_5_Z**
M3D_BASE+6DCh|EDCh	Spotlight Exponent for Light Source 5 Register	**M3D_S_RL_5**
M3D_BASE+6E0h|EE0h	X Direction of Spotlight for Light Source 6 Register	**M3D_S_DL_6_X**
M3D_BASE+6E4h|EE4h	Y Direction of Spotlight for Light Source 6 Register	**M3D_S_DL_6_Y**
M3D_BASE+6E8h|EE8h	Z Direction of Spotlight for Light Source 6 Register	**M3D_S_DL_6_Z**
M3D_BASE+6ECh|EECh	Spotlight Exponent for Light Source 6 Register	**M3D_S_RL_6**
M3D_BASE+6F0h|EF0h	X Direction of Spotlight for Light Source 7 Register	**M3D_S_DL_7_X**
M3D_BASE+6F4h|EF4h	Y Direction of Spotlight for Light Source 7 Register	**M3D_S_DL_7_Y**
M3D_BASE+6F8h|EF8h	Z Direction of Spotlight for Light Source 7 Register	**M3D_S_DL_7_Z**
M3D_BASE+6FCh|EFCh	Spotlight Exponent for Light Source 7 Register	**M3D_S_RL_7**
M3D_BASE+700h|F00h	Spotlight Cutoff Angle for Light Source 0 Register	**M3D_C_RL_0**
M3D_BASE+704h|F04h	Constant Attenuation Factor for Light Source 0 Register	**M3D_K_0_0**
M3D_BASE+708h|F08h	Linear Attenuation Factor for Light Source 0 Register	**M3D_K_1_0**
M3D_BASE+70Ch|F0Ch	Quadratic Attenuation Factor for Light Source 0 Register	**M3D_K_2_0**
M3D_BASE+710h|F10h	Spotlight Cutoff Angle for Light Source 1 Register	**M3D_C_RL_1**
M3D_BASE+714h|F14h	Constant Attenuation Factor for Light Source 1 Register	**M3D_K_0_1**
M3D_BASE+718h|F18h	Linear Attenuation Factor for Light Source 1 Register	**M3D_K_1_1**
M3D_BASE+71Ch|F1Ch	Quadratic Attenuation Factor for Light Source 1 Register	**M3D_K_2_1**
M3D_BASE+720h|F20h	Spotlight Cutoff Angle for Light Source 2 Register	**M3D_C_RL_2**
M3D_BASE+724h|F24h	Constant Attenuation Factor for Light Source 2 Register	**M3D_K_0_2**
M3D_BASE+728h|F28h	Linear Attenuation Factor for Light Source 2 Register	**M3D_K_1_2**
M3D_BASE+72Ch|F2Ch	Quadratic Attenuation Factor for Light Source 2 Register	**M3D_K_2_2**

M3D_BASE+730h|F30h	Spotlight Cutoff Angle for Light Source 3 Register	**M3D_C_RL_3**
M3D_BASE+734h|F34h	Constant Attenuation Factor for Light Source 3 Register	**M3D_K_0_3**
M3D_BASE+738h|F38h	Linear Attenuation Factor for Light Source 3 Register	**M3D_K_1_3**
M3D_BASE+73Ch|F3Ch	Quadratic Attenuation Factor for Light Source 3 Register	**M3D_K_2_3**
M3D_BASE+740h|F40h	Spotlight Cutoff Angle for Light Source 4 Register	**M3D_C_RL_4**
M3D_BASE+744h|F44h	Constant Attenuation Factor for Light Source 4 Register	**M3D_K_0_4**
M3D_BASE+748h|F48h	Linear Attenuation Factor for Light Source 4 Register	**M3D_K_1_4**
M3D_BASE+74Ch|F4Ch	Quadratic Attenuation Factor for Light Source 4 Register	**M3D_K_2_4**
M3D_BASE+750h|F50h	Spotlight Cutoff Angle for Light Source 5 Register	**M3D_C_RL_5**
M3D_BASE+754h|F54h	Constant Attenuation Factor for Light Source 5 Register	**M3D_K_0_5**
M3D_BASE+758h|F58h	Linear Attenuation Factor for Light Source 5 Register	**M3D_K_1_5**
M3D_BASE+75Ch|F5Ch	Quadratic Attenuation Factor for Light Source 5 Register	**M3D_K_2_5**
M3D_BASE+760h|F60h	Spotlight Cutoff Angle for Light Source 6 Register	**M3D_C_RL_6**
M3D_BASE+764h|F64h	Constant Attenuation Factor for Light Source 6 Register	**M3D_K_0_6**
M3D_BASE+768h|F68h	Linear Attenuation Factor for Light Source 6 Register	**M3D_K_1_6**
M3D_BASE+76Ch|F6Ch	Quadratic Attenuation Factor for Light Source 6 Register	**M3D_K_2_6**
M3D_BASE+770h|F70h	Spotlight Cutoff Angle for Light Source 7 Register	**M3D_C_RL_7**
M3D_BASE+774h|F74h	Constant Attenuation Factor for Light Source 7 Register	**M3D_K_0_7**
M3D_BASE+778h|F78h	Linear Attenuation Factor for Light Source 7 Register	**M3D_K_1_7**
M3D_BASE+77Ch|F7Ch	Quadratic Attenuation Factor for Light Source 7 Register	**M3D_K_2_7**
M3D_BASE+780h|F80h	Red Component of Ambient Color of Scene Register	**M3D_A_Cs_R**
M3D_BASE+784h|F84h	Green Component of Ambient Color of Scene Register	**M3D_A_CS_G**

© 2009 MediaTek Inc.

M3D_BASE+788h|F88h	Blue Component of Ambient Color of Scene Register	**M3D_A_CS_B**
M3D_BASE+78Ch	Color cache hit counter	**M3D_CC_CNT**
M3D_BASE+790h|F90h	Specular Exponent of Material Register	**M3D_S_RM**
M3D_BASE+794h|F94h	Fog Density Register	**M3D_FOG_DENSITY**
M3D_BASE+798h|F98h	Fog Start Register	**M3D_FOG_START**
M3D_BASE+79Ch|F9Ch	Fog End Register	**M3D_FOG_END**
M3D_BASE+7A0h|FA0h	Lower Bound of Point Size Register	**M3D_POINT_SIZE_MIN**
M3D_BASE+7A4h|FA4h	Upper Bound of Point Size Register	**M3D_PONIT_SIZE_MAX**
M3D_BASE+7A8h|FA8h	Constant Coefficient of Distance Attenaution Fucntion for Point Size Register	**M3D_POINT_ATTENUATION_A**
M3D_BASE+7ACh|FACh	Linear Coefficient of Distance Attenaution Fucntion for Point Size Register	**M3D_POINT_ATTENUATION_B**
M3D_BASE+7B0h|FB0h	Quadratic Coefficient of Distance Attenaution Fucntion for Point Size Register	**M3D_POINT_ATTENUATION_C**
M3D_BASE+7B4h	M3D texture #0 filter/wrap mode	**M3D_TEX_PARA_0**
M3D_BASE+7B8h	M3D texture #1 filter/wrap mode	**M3D_TEX_PARA_1**
M3D_BASE+7BCh	M3D texture #2 filter/wrap mode	**M3D_TEX_PARA_2**
M3D_BASE+7C0h	M3D cache performance counter control	**M3D_CACHE_PERF_CTRL**
M3D_BASE+7C4h	Reserved	**Reserved**
M3D_BASE+7C8h	M3D drawtex TEX0 cropper region u coord	**M3D_DRAWTEX_CRU_0**
M3D_BASE+7CCh	M3D drawtex TEX0 cropper region v coord	**M3D_DRAWTEX_CRV_0**
M3D_BASE+7D0h	M3D drawtex TEX0 cropper region delta u coord	**M3D_DRAWTEX_DCRU_0**
M3D_BASE+7D4h	M3D drawtex TEX0 cropper region delta v coord	**M3D_DRAWTEX_DCRV_0**
M3D_BASE+7D8h	M3D drawtex cropper region xy coord	**M3D_DRAWTEX_XY**
M3D_BASE+7DCh	M3D drawtex cropper region width/height	**M3D_DRAWTEX_WH**
M3D_BASE+7E0h	M3D drawtex cropper region z value	**M3D_DRAWTEX_Z**
M3D_BASE+7E4h	M3D drawtex cropper region fog factor	**M3D_DRAWTEX_FOG**
M3D_BASE+7E8h	M3D dummy register 0	**M3D_DUMMY_0**
M3D_BASE+7ECh	M3D dummy register 1	**M3D_DUMMY_1**
M3D_BASE+7F0h	M3D dummy register 2	**M3D_DUMMY_2**
M3D_BASE+7F4h	M3D debug port 5	**M3D_DBGRD_5**
M3D_BASE+7F8h	M3D debug port 6	**M3D_DBGRD_6**

© 2009 MediaTek Inc.

M3D_BASE+7FCh	M3D debug port 7	**M3D_DBGRD_7**

Table 118 ***M3D Registers***

M3D_BASE+000h M3D trigger register M3D_TRIGGER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												M3D_DRAWTEX_ENABLE	RESERVED	RESERVED	RESERVED	M3D_TRIGGER
Type												R/W				R/W
Reset												0				0

M3D_TRIGGER Trigger the M3D hardware pipeline. It CANNOT be enabled when buffer clear operation is running.

M3D_DRAWTEX_ENABLE Trigger drawtex function

When hardware operation is done, the corresponding bit would be cleared by hardware automatically.

M3D_BASE+004h M3D reset register M3D_RESET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																RESET
Type																R/W
Reset																0

RESET .Reset M3D hardware pipeline.

M3D_BASE+008h M3D status register M3D_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										M3D_COLOR_FLUSH_BUSY	M3D_Z_FLUSH_BUSY	RESERVED	RESERVED	RESERVED	RESERVED	M3D_BUSY
Type										RO	RO					RO
Reset										0	0					0

 © 2009 MediaTek Inc.

M3D_BUSY Indicating that M3D hardware is busy.
M3D_Z__FLUSH_BUSY Indicating that M3D is busy in z cache flushing.
M3D_COLOR_FLUSH_BUSY Indicating that M3D is busy in color cache flushing.

M3D_BASE+00Ch M3D interrupt enable register M3D_INTEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										M3D_COLOR_FLUSH_DONE_INTEN	M3D_Z_FLUSH_DONE_INTEN	RESERVED	RESERVED	RESERVED	RESERVED	M3D_RENDER_DONE_INTEN
Type										R/W	R/W					R/W
Reset										0	0					0

M3D_RENDER_DONE_INTEN Enable interrupting when rendering is done.
M3D_Z_FLUSH_DONE_INTEN Enable interrupting when z cache flush is done.
M3D_COLOR_FLUSH_DONE_INTEN Enable interrupting when color cache flush is done.

M3D_BASE+010h M3D interrupt status register M3D_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										M3D_COLOR_FLUSH_DONE_INTSTA	M3D_Z_FLUSH_DONE_INTSTA	RESERVED	RESERVED	RESERVED	RESERVED	M3D_RENDER_DONE_INTSTA
Type										WC	WC					WC
Reset										0	0					0

M3D_RENDER_DONE_INTSTA Interrupting status of rendering.
M3D_Z_FLUSH_DONE_INTSTA Interrupting status of z cache flush.
M3D_COLOR_FLUSH_DONE_INTSTA Interrupting status of color cache flush.

M3D_BASE+014h M3D fragment cache enable register M3D_FRAG_CACHE_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													M3D_TEX_CACHE_ENABLE	M3D_COLORL_CACHE_ENABLE	M3D_Z_CACHE_ENABLE	M3D_STENCIL_CACHE_ENABLE
Type													R/W	R/W	R/W	R/W
Reset													1	1	1	1

M3D_STENCIL_CACHE_ENABLE Enable M3D stencil cache.
M3D_Z_CACHE_ENABLE Enable M3D z cache.
M3D_COLOR_CACHE_ENABLE Enable M3D color cache.
M3D_TEX_CACHE_ENABLE Enable M3D tex cache.

M3D_BASE+018h M3D fragment cache flush register M3D_FRAG_CACHE_FLUSH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														M3D_COLORL_CACHE_FLUSH	M3D_Z_CACHE_FLUSH	M3D_STENCIL_CACHE_FLUSH
Type														R/W	R/W	R/W
Reset														0	0	0

M3D_STENCIL_CACHE_FLUSH Flush the content of M3D stencil cache.
M3D_Z_CACHE_FLUSH Flush the content of M3D z cache.
M3D_COLOR_CACHE_FLUSH Flush the content of M3D color cache.

M3D_BASE+01Ch M3D fragment cache invalidate register M3D_FRAG_CACHE_INVAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														M3D_COLORL_CACHE_INVAL	M3D_Z_CACHE_INVLA	M3D_STENCIL_CACHE_INVLAL
Type														R/W	R/W	R/W
Reset														0	0	0

M3D_STENCIL_CACHE_INVAL Invalidate all cache lines in M3D stencil cache.
M3D_Z_CACHE_INVAL Invalidate all cache lines in M3D z cache.

M3D_COLOR_CACHE_INVAL Invalidate all cache lines in M3D color cache.

M3D_BASE+020h M3D fragment cache reset register M3D_FRAG_CACHE_RESET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														M3D_COLOR_CACHE_RESET	M3D_Z_CACHE_RESET	M3D_STECIL_CACHE_RESET
Type														R/W	R/W	R/W
Reset														0	0	0

M3D_STENCIL_CACHE_RESET Reset the M3D stencil cache hardware state machine.
M3D_Z_CACHE_RESET Reset the M3D z cache hardware state machine.
M3D_COLOR_CACHE_RESET Reset m3d stencil cache hardware state machine

M3D_BASE+024h M3D texture cache clear register M3D_TEX_CACHE_CLEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_TEX_CACHE_CLEAR
Type																R/W
Reset																0

M3D_TEX_CACHE_CLEAR Clear texture cache's valid bits and counter state.

M3D_BASE+028h M3D primitive mode register M3D_PRIMITIVE _MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													M3D_PRIMITIVE _MODE			
Type													R/W			
Reset													0			

M3D_PRIMITIVE_MODE Primitive mode.
0x0 represent points

0x4 represent lines
0x6 represent line loop
0x7 represent line strip
0x8 represent triangles
0xA represent triangle strip 0
0xB represent triangle strip 1
0xC represent triangle fan 0
0xD represent triangle fan 1

Primitive Type	OpenGL-ES	D3DM
Points	0x0	
Lines	0x4	
Line Loop	0x6	-
Line Strip	0x7	
Triangles	0x8	
Triangle Strip	0xA	0xB
Triangle Fan	0xC	0xD

M3D_BASE+02Ch M3D draw mode register M3D_PRIMITIVE_DRAW_MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_PRIMITIVE_DRAW_MODE
Type																R/W

M3D_PRIMITIVE_DRAW_MODE Draw mode. (0 represent draw-array, 1 represent draw-element)

M3D_BASE+030h M3D vertex count register M3D_PRIMITIVE_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PRIMITIVE_COUNT															
Type	R/W															

M3D_PRIMITIVE_COUNT Specifies the number of vertices to be rendered.

M3D_BASE+034h M3D draw-array first index register M3D_DRAW_ARRAY_FIRST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DRAW_ARRAY_FIRST															
Type	R/W															

M3D_DRAW_ARRAY_FIRST Specifies the starting index of draw-array.

M3D_BASE+038h M3D draw-array end index register M3D_DRAW_ARRAY_END

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DRAW_ARRAY_END															
Type	R/W															

M3D_DRAW_ARRAY_END Specifies the end index of draw-array.

M3D_BASE+03Ch M3D draw-element index type register M3D_DRAW_ELEMENT_TYPE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													M3D_DRAW_ELEMENT_TYPE			
Type													R/W			

M3D_DRAW_ELEMENT_TYPE Specifies the data type of each index in the array. (0x1 represent unsigned byte, 0x3 represent unsigned short)

M3D_BASE+040h M3D draw-element pointer register M3D_DRAW_ELEMENT_POINTER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DRAW_ELEMENT_POINTER[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DRAW_ELEMENT_POINTER[15:0]															
Type	R/W															

M3D_DRAW_ELEMENT_POINTER Specifies the memory address of the index array for deaw-element. 8 bytes alignment.

M3D_BASE+044h M3D primitive anti-aliasing register M3D_PRIMITIVE_AA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														M3D_POINT_SMOOTH	M3D_LINE_SMOOTH	
Type														R/W	R/W	
Reset														0	0	

M3D_LINE_SMOOTH Line smooth disable/enable. (0 represent disable, 1 represent enable)
M3D_POINT_SMOOTH Point smooth disable/enable. (0 represent disable, 1 represent enable)
Driver should do some judgements before enable point smooth.

- POINT_SMOOTH is enable
- No glEnableClientState(GL_POINT_SIZE_ARRAY_OES)
- glPointSizex(fixed size), size <= 4 and POINT_DISTANCE_ATTENUATION is (1, 0, 0)

M3D_BASE+048h M3D polygon offset enable/disable register M3D_POLYGON_OFFSET_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_POLYGON_OFFSET_ENABLE
Type																R/W
Reset																0

M3D_POLYGON_OFFSET_ENABLE Polygon offset disable/enable. (0 represent disable, 1 represent enable)

M3D_BASE+04Ch|84Ch M3D polygon offset factor register M3D_POLYGON_OFFSET_FACTOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_POLYGON_OFFSET_FACTOR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_POLYGON_OFFSET_ FACTOR [15:0]															
Type	R/W															

M3D_POLYGON_OFFSET_FACTOR Specifies a scale factor that is used to create a variable depth offset for each polygon.

M3D_BASE+050h|850h M3D polygon offset unit register M3D_POLYGON_OFFSET_UNITS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_POLYGON_OFFSET_UNITS[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_POLYGON_OFFSET_UNITS[15:0]															
Type	R/W															

M3D_POLYGON_OFFSET_UNITS Original units is multiplied by an implementation-specific value to create a constant depth offset. (r * units)

M3D_BASE+054h|854h M3D line width register M3D_LINE_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_LINE_WIDTH[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_LINE_WIDTH[15:0]															
Type	R/W															

M3D_LINE_WIDTH Specify the width of rasterized lines.

M3D_BASE+058h M3D point-sprite register M3D_POINT_SPRITE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													M3D_POINT_COORD_REPLACE			M3D_POINT_SPRITE_ENABLE
Type												R/W				R/W
Reset												0				0

M3D_POINT_SPRITE_ENABLE Point-sprite disable/enable. (0 represent disable, 1 represent enable)

M3D_POINT_COORD_REPLACE Replace original texture coordinate with new when point-sprite is enable. (0 represent disable, 1 represent enable)

M3D_BASE+05Ch M3D vertex array register M3D_VERTEX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										M3D_VERTEX_SCREEN	M3D_VERTEX_TYPE				M3D_VERTEX_SIZE	
Type										R/W	R/W				R/W	
Reset										0	0				0	

M3D_VERTEX_SIZE Specifies the number of coordinates per vertex. (0 represent 2 elements, 1 represent 3 elements, 2 represent 4 elements)

M3D_VERTEX_TYPE Specifies the data type of each vertex coordinate in the array. (0x0 represent byte, 0x2 represent short, 0x6 represent float, 0xc represent fixed)

M3D_VERTEX_SCREEN Specifies the vertex coordinate is screnn space coordinate or not. (0 represent object space, 1 represent screen space : x, y, z, rhw)

M3D_BASE+060h M3D vertex array stride register M3D_VERTEX_STRIDE_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VERTEX_STRIDE_B[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VERTEX_STRIDE_B[15:0]															
Type	R/W															

M3D_VERTEX_STRIDE_B Specifies the byte offset between consecutive vertices.

M3D_BASE+064h M3D vertex array pointer register M3D_VERTEX_POINTER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VERTEX_POINTER[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VERTEX_POINTER[15:0]															
Type	R/W															

M3D_VERTEX_POINTER Specifies a pointer to the first coordinate of the first vertex in the array.

M3D_BASE+068h M3D vertex cache pointer register M3D_VERTEX_CACHE_POINTER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VERTEX_CACHE_POINTER[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VERTEX_CACHE_POINTER[15:0]															
Type	R/W															

M3D_VERTEX_CACHE_POINTER Vertex cache pointer.
Vertex cache must use internal memory (GMC2 : 0x4002_0000 ~ 0x4004_3FFF). 8 bytes alignment. Need total 2048 bytes (32 entries, each entry has 64 bytes).

M3D_BASE+06Ch M3D bounding box control register M3D_BBOX_EXPAND

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															M3D_BBOX_EXPAND_Y_EN	M3D_BBOX_EXPAND_X_EN
Type															R/W	R/W
Reset															1	1

M3D_BBOX_EXPAND_X_EN Expand bounding box in x direction. (0 represent disable, 1 represent enable)
M3D_BBOX_EXPAND_Y_EN Expand bounding box in y direction. (0 represent disable, 1 represent enable)
Default is enable this function. Please disable this function when viewport isn't equal to screen.

M3D_BASE+070h M3D normal array type register M3D_NORMAL_TYPE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													M3D_NORMAL_TYPE			
Type													R/W			

M3D_NORMAL_TYPE Specifies the data type of each coordinate in the array. (0x0 represent byte, 0x2 represent short, 0x6 represent float, 0xc represent fixed)

M3D_BASE+074h M3D normal array stride register M3D_NORMAL_STRIDE_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_STRIDE_B[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_STRIDE_B[15:0]															
Type	R/W															

M3D_NORMAL_STRIDE_B Specifies the byte offset between consecutive normals.

M3D_BASE+078h M3D normal array pointer register M3D_NORMAL_POINTER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_POINTER[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_POINTER[15:0]															
Type	R/W															

 © 2009 MediaTek Inc.

M3D_NORMAL_POINTER Specifies a pointer to the first coordinate of the first normal in the array.

M3D_BASE+07Ch M3D color array type/size/input register M3D_COLOR_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									M3D_COLOR_TYPE_0				M3D_COLOR_SIZE_0	M3D_COLOR_ORDER_0	M3D_COLOR_INPUT_0	M3D_COLOR_ENABLE_0
Type									R/W				R/W	R/W	R/W	R/W
Reset									0				0	0	0	1

M3D_COLOR_ENABLE_0 Specifies the color disable/enable. (0 represent disable, 1 represent enable)

M3D_COLOR_INPUT_0 Specifies the source of color input. (0 represent register, 1 represent array)

M3D_COLOR_ORDER_0 Specifies the color order. (0x0 represent ABGR, 0x1 represent ARGB)

M3D_COLOR_SIZE_0 Specifies the number of coordinates per color. (0 represent 3 elements, 1 represent 4 elements)

M3D_COLOR_TYPE _0 Specifies the data type of each coordinate in the array. (0x1 represent ubyte, 0x6 represent float, 0xc represent fixed)

M3D_BASE+080h M3D color array type/size/input register M3D_COLOR_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									M3D_COLOR_TYPE_1				M3D_COLOR_SIZE_1	M3D_COLOR_ORDER_1	_1M3D_COLOR_INPUT_1	M3D_COLOR_ENABLE_1
Type									R/W				R/W	R/W	R/W	R/W
Reset									0				0	0	0	1

M3D_COLOR_ENABLE_1 Specifies the color disable/enable. (0 represent disable, 1 represent enable)

M3D_COLOR_INPUT_1 Specifies the source of color input. (0 represent register, 1 represent array)

M3D_COLOR_ORDER_1 Specifies the color order. (0x0 represent ABGR, 0x1 represent ARGB)

M3D_COLOR_SIZE_1 Specifies the number of coordinates per color. (0 represent 3 elements, 1 represent 4 elements)

M3D_COLOR_TYPE _1 Specifies the data type of each coordinate in the array. (0x1 represent ubyte, 0x6 represent float, 0xc represent fixed)

M3D_BASE+084h M3D color array stride register M3D_COLOR_STRIDE_B_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_STRIDE_B_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_STRIDE_B_0[15:0]															
Type	R/W															

M3D_COLOR_STRIDE_B_0 Specifies the byte offset between consecutive colors.

M3D_BASE+088h M3D color array stride register M3D_COLOR_STRIDE_B_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_STRIDE_B_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_STRIDE_B_1[15:0]															
Type	R/W															

M3D_COLOR_STRIDE_B_1 Specifies the byte offset between consecutive colors.

M3D_BASE+08Ch M3D color array pointer register M3D_COLOR_POINTER_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_POINTER_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_POINTER_0[15:0]															
Type	R/W															

M3D_COLOR_POINTER_0 Specifies a pointer to the first coordinate of the first color in the array.

M3D_BASE+090h M3D color array pointer register M3D_COLOR_POINTER_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_POINTER_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_POINTER_1[15:0]															
Type	R/W															

M3D_COLOR_POINTER_1 Specifies a pointer to the first coordinate of the first color in the array.

M3D_BASE+094h M3D debug port 0 M3D_DBGRD_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	M3D_DBGRD_0[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_0[15:0]															
Type	R															

M3D_DBGRD_0 M3D engine debug port #0 (read only).

M3D_BASE+098h M3D debug port 1 M3D_DBGRD_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_1[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_1[15:0]															
Type	R															

M3D_DBGRD_1 M3D engine debug port #1 (read only).

M3D_BASE+09Ch M3D debug port 2 M3D_DBGRD_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_2[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_2[15:0]															
Type	R															

M3D_DBGRD_2 M3D engine debug port #2 (read only).

M3D_BASE+0A0h M3D debug port 3 M3D_DBGRD_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_3[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_3[15:0]															
Type	R															

M3D_DBGRD_3 M3D engine debug port #3 (read only).

M3D_BASE+0A4h M3D debug port 4 M3D_DBGRD_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_4[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_4[15:0]															
Type	R															

M3D_DBGRD_4 M3D engine debug port #4 (read only).

© 2009 MediaTek Inc.

M3D_BASE+0A8h M3D texture coordinate 0 register M3D_TEX_COORD_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				M3D_TEX_COORD_TYPE_0				M3D_TEX_COORD_SIZE_0			M3D_TEX_COORD_XFROM_PROJ_0	M3D_TEX_COORD_XFORM_ENABLE_0	M3D_TEX_COORD_WRAP_T_0	M3D_TEX_COORD_WRAP_S_0	M3D_TEX_COORD_2D_0	M3D_TEX_COORD_ENABLE_0
Type				R/W				R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset				0				0			1	0	0	0	1	0

M3D_TEX_COORD_ENABLE_0 Tetxure coordinate disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_2D_0 Specifies the dimension of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_S_0 Specifies the s wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_T_0 Specifies the t wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_XFORM_ENABLE_0 Texture coordinate transform disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_XFORM_PROJ_0 Texture coordinate transform projected disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_SIZE_0 Specifies the number of coordinates per array element.

0x0 represent 1 element. (S, 1, 0, 0)
0x2 represent 2 elements. (S, T, 0, 1)
0x3 represent 2 elements. (S, T, 1, 0)
0x4 represent 3 elements. (S, T, R, 1)
0x6 represent 4 elements. (S, T, R, Q)

Size	OpenGL-ES	D3DM
1	-	0x0
2	0x2	0x3
3	0x4	
4	0x6	-

M3D_TEX_COORD_TYPE_0 Specifies the type of each texture coordinate.(0x0 represent byte, 0x2 represent short, 0x6 represent float, 0xc represent fixed)

M3D_BASE+0ACh **M3D texture coordinate 0 register** **M3D_TEX_COORD_1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				M3D_TEX_COORD_TYPE_1				M3D_TEX_COORD_SIZE_1			M3D_TEX_COORD_XFROM_PROJ_1	M3D_TEX_COORD_XFORM_ENABLE_1	M3D_TEX_COORD_WRAP_T_1	M3D_TEX_COORD_WRAP_S_1	M3D_TEX_COORD_2D_1	M3D_TEX_COORD_ENABLE_1
Type				R/W				R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset				0				0			1	0	0	0	1	0

M3D_TEX_COORD_ENABLE_1 Tetxure coordinate disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_2D_1 Specifies the dimension of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_S_1 Specifies the s wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_T_1 Specifies the t wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_XFORM_ENABLE_1 Texture coordinate transform disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_XFORM_PROJ_1 Texture coordinate transform projected disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_SIZE_1 Specifies the number of coordinates per array element.

0x0 represent 1 element. (S, 1, 0, 0)
0x2 represent 2 elements. (S, T, 0, 1)
0x3 represent 2 elements. (S, T, 1, 0)
0x4 represent 3 elements. (S, T, R, 1)
0x6 represent 4 elements. (S, T, R, Q)

Size	OpenGL-ES	D3DM
1	-	0x0
2	0x2	0x3
3	0x4	
4	0x6	-

M3D_TEX_COORD_TYPE_1 Specifies the type of each texture coordinate.(0x0 represent byte, 0x2 represent short, 0x6 represent float, 0xc represent fixed)

M3D_BASE+0B0h M3D texture coordinate 0 register M3D_TEX_COORD_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				M3D_TEX_COORD_TYPE_2				M3D_TEX_COORD_SIZE_2			M3D_TEX_COORD_XFROM_PROJ_2	M3D_TEX_COORD_XFORM_ENABLE_2	M3D_TEX_COORD_WRAP_T_0	M3D_TEX_COORD_WRAP_S_2	M3D_TEX_COORD_2D_2	M3D_TEX_COORD_ENABLE_2
Type				R/W				R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset				0				0			1	0	0	0	1	0

M3D_TEX_COORD_ENABLE_2 Tetxure coordinate disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_2D_2 Specifies the dimension of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_S_2 Specifies the s wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_WRAP_T_2 Specifies the t wrapping of each tetxure coordinate. (0x0 represent 1D, 0x1 represent 2D)

M3D_TEX_COORD_XFORM_ENABLE_2 Texture coordinate transform disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_XFORM_PROJ_2 Texture coordinate transform projected disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_COORD_SIZE_2 Specifies the number of coordinates per array element.

0x0 represent 1 element. (S, 1, 0, 0)
0x2 represent 2 elements. (S, T, 0, 1)
0x3 represent 2 elements. (S, T, 1, 0)
0x4 represent 3 elements. (S, T, R, 1)
0x6 represent 4 elements. (S, T, R, Q)

Size	OpenGL-ES	D3DM
1	-	0x0
2	0x2	0x3
3	0x4	
4	0x6	-

M3D_TEX_COORD_TYPE_2 Specifies the type of each texture coordinate.(0x0 represent byte, 0x2 represent short, 0x6 represent float, 0xc represent fixed)

M3D_BASE+0B4h M3D texture array 0 stride register M3D_TEX_COORD_STRIDE_B_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_STRIDE_B_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_STRIDE_B_0[15:0]															
Type	R/W															

M3D_TEX_COORD_STRIDE_B_0 Specifies the byte offset between consecutive elements.

M3D_BASE+0B8h M3D texture array 1 stride register M3D_TEX_COORD_STRIDE_B_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_STRIDE_B_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_STRIDE_B_1[15:0]															
Type	R/W															

M3D_TEX_COORD_STRIDE_B_1 Specifies the byte offset between consecutive elements.

M3D_BASE+0BCh M3D texture array 2 stride register M3D_TEX_COORD_STRIDE_B_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_STRIDE_B_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_STRIDE_B_2[15:0]															
Type	R/W															

M3D_TEX_COORD_STRIDE_B_2 Specifies the byte offset between consecutive elements.

M3D_BASE+0C0h M3D texture array 0 pointer register M3D_TEX_COORD_POINTER_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_POINTER_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_POINTER_0[15:0]															
Type	R/W															

M3D_TEX_COORD_POINTER_0 Specifies a pointer to the first coordinate of the first element in the array.

 © 2009 MediaTek Inc.

M3D_BASE+0C4h M3D texture array 1 pointer register M3D_TEX_COORD_POINTER_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_POINTER_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_POINTER_1 [15:0]															
Type	R/W															

M3D_ TEX_COORD_POINTER_1 Specifies a pointer to the first coordinate of the first element in the array.

M3D_BASE+0C8h M3D texture array 2 pointer register M3D_TEX_COORD_POINTER_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_COORD_POINTER_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_COORD_POINTER_2[15:0]															
Type	R/W															

M3D_ TEX_COORD_POINTER_2 Specifies a pointer to the first coordinate of the first element in the array.

M3D_BASE+0CCh M3D point array type/input register M3D_PNT_SIZE_INPUT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												M3D_PNT_SIZE_TYPE				M3D_PNT_SIZE_INPUT
Type												R/W				R/W
Reset												0				0

M3D_PNT_SIZE_INPUT Specifies the source of point size. (0 represent register, 1 represent array)

M3D_PNT_SIZE_TYPE Specifies the type of point size. (0x06 represent float , 0x0c represent fixed)

M3D_BASE+0D0h|8D0h M3D point array size register M3D_PNT_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PNT_SIZE[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PNT_SIZE[15:0]															
Type	R/W															

M3D_PNT_SIZE Set the current point size. (00D0h : data format is s15.16, 20D0h : data format is s[8].23)

M3D_BASE+0D4h M3D point array stride register M3D_PNT_SIZE_STRIDE_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PNT_SIZE_STRIDE_B[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PNT_SIZE_STRIDE_B[15:0]															
Type	R/W															

M3D_PNT_SIZE_STRIDE_B Specifies the byte offset between consecutive point size.

M3D_BASE+0D8h M3D point array pointer register M3D_PNT_SIZE_POINTER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PNT_SIZE_POINTER[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PNT_SIZE_POINTER[15:0]															
Type	R/W															

M3D_PNT_SIZE_POINTER Specifies a pointer to the first point size in the array.

M3D_BASE+0DCh M3D projection matrix type register M3D_PROJ_MATRIX_TYPE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															M3D_PROJ_MATRIX_TYPE	
Type															R/W	
Reset															0	

M3D_PROJ_MATRIX_TYPE Projection matrix type. (0 represent general, 1 represent frustum, 2 represent ortho)

M3D_BASE+0E0h M3D user-defined clipping plane enable/disable register M3D_CLIP_PLANE_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name																M3D_CLIP_PLANE_ENABLE
Type																R/W
Reset																0

M3D_CLIP_PLANE_ENABLE User-defined clipping plane disable/enable. (0 represent disable, 1 represent enable)

M3D_BASE+0E4h M3D normal scale enable/disable register M3D_NORMAL_SCALE_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_NORMAL_SCALE_ENABLE
Type																R/W
Reset																0

M3D_NORMAL_SCALE_ENABLE Normal scale disable/enable. (0 represent disable, 1 represent enable)

M3D_BASE+0E8h M3D line last pixel M3D_LINE_LAST_PIXEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_LINE_LAST_PIXEL
Type																R/W
Reset																0

M3D_LINE_LAST_PIXEL Line last pixel disable/enable. (0 represent disable, 1 represent enable)

M3D_BASE+0ECh M3D primitive culling register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									M3D_FRONT_FACE	M3D_CULL_FACE		M3D_CULL_CLIP_PLANE	M3D_CULL_Z	M3D_CULL_ZERO_AREA	M3D_CULL_ALL_OUT	M3D_CULL_SMALL_TRIANGLE
Type									R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset									1	1		1	1	1	1	1

M3D_CULL_SMALL_TRIANGLE Cull triangle that don't enclose any pixel center. (0 represent disable, 1 represent enabl)

M3D_CULL_ALL_OUT Trivial rejection when all vertices of a primitive is clipped. (0 represent disable, 1 represent enable)

M3D_CULL_ZERO_AREA Trivial rejection when the area of a primitive is zero. (0 represent disable, 1 represent enable)

M3D_CULL_Z Z culling. (0 represent disable, 1 represent enable)

M3D_CULL_CLIP_PLANE_0 User-defined clipping plane culling. (0 represent disable, 1 represent enable)

M3D_CULL_FACE Specifies whether front- or back-facing polygons are culled. (0 represent front, 1 represent back, 2 represent front and back, 3 represent none)

M3D_FRONT_FACE Specifies the orientation of front-facing polygons. (0 represent CW, 1 represent CCW)

M3D_BASE+0F0h M3D shading model register M3D_SHADE_MODEL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															M3D_FLAT_TYPE	M3D_SHADE_MODEL
Type															R/W	R/W
Reset															0	1

M3D_SHADE_MODEL Shading model. (0 represent flat, 1 represent smooth)

M3D_FLAT_TYPE Flat shading type. (0 represent the last vertex, 1 represent the first vertex)

M3D_BASE+0F4h M3D Lighting Control Register M3D_LIGHT_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		M3D_LIGHT_INF_POS	M3D_LIGHT_F_ENABLE	M3D_FOG_CUSTOM	M3D_LIGHT_RANGE_ENBALE	M3D_LIGHT_LVIEW_ENABLE	M3D_LIGHT_SPOT_ENABLE	M3D_COLOR_SUM_LIGHT			M3D_COLOR_MATERIAL_SPECULAR		M3D_COLOR_MATERIAL_DIFFUSE		M3D_COLOR_MATERIAL_AMBIENT	
Type		R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W		R/W		R/W	
Reset		0	0	0	0	0	0	0			0		0		0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			M3D_FOG_MODE		M3D_FOG_ENABLE	M3D_NORMALIZE		M3D_LIGHT_7_ENABLE	M3D_LIGHT_6_ENABLE	M3D_LIGHT_5_ENABLE	M3D_LIGHT_4_ENABLE	M3D_LIGHT_3_ENABLE	M3D_LIGHT_2_ENABLE	M3D_LIGHT_1_ENABLE	M3D_LIGHT_0_ENABLE	M3D_LIGHTING_ENABLE
Type			R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0		0	0		0	0	0	0	0	0	0	0	0

M3D_LIGHTING_ENABLE Enable global lighting and thus m3d_lightx_enable is valid.

M3D_LIGHT_x_ENABLE Enable Lightx, where x is 0 ~ 7, i.e., supporting 8 light sources at most.

M3D_NORMALIZE Enable normalization of vertex normal which is transformed in TnL stage.

M3D_FOG_ENABLE Enable fog. When it is enabled, the fog factor is calculated in lighting engine and passed to next stage.

M3D_FOG_MODE Fog mode.

- **00** EXP.
- **01** EXP2.
- **10** LINEAR.
- **11** Reserved.

M3D_COLOR_MATERIAL_AMBIENT Enable color material for ambient item. In OpenGL-ES, when color material is set, the two bits should be set to ‘01’ to select color from vertex color 1.

- **00** Material color.
- **01** From Color 1.
- **10** From Color 2.
- **11** Reserved.

M3D_COLOR_MATERIAL_DIFFUSE Enable color material for diffuse item. In OpenGL-ES, when color material is set, the two bits should be set to ‘01’ to select color from vertex color 1.

- **00** Material color.
- **01** From Color 1.
- **10** From Color 2.
- **11** Reserved.

M3D_COLOR_MATERIAL_SPECULAR Enable color material for specular item.

- **00** Material color.
- **01** From Color 1.
- **10** From Color 2.
- **11** Reserved.

M3D_COLOR_SUM_LIGHT To indicate if the two colors of one vertex should be summed or not. In OpenGL-ES, the diffuse color and specular color are summed in light engine, so it must be set to 1. In D3DM, it must be set to 0.

- **0** Not summed.
- **1** Summed.

M3D_LIGHT_SPOT_ENABLE To enable/disable spot light source. In D3DM, it does not support spot light source, so the bit must be set to 0. In OpenGL-ES, it must be set to 1.

- **0** Disable.
- **1** Enable.

M3D_LIGHT_LVIEW_ENABLE To indicate if local view is supported or not.

- **0** Disable.
- **1** Enable.

M3D_LIGHT_RANGE_EANBLE If this bit is enabled, then the attenuation item is zero while the distance between the vertex and the light source exceeds light's range.

- **0** Disable.
- **1** Enable.

M3D_FOG_CUTOM To determine the fog factor is calculated by fog factor equation or is assigned through alpha value of the second color.

- **0** Calculated by fog factor equation.
- **1** Assigned through alpha value of the second color.

M3D_LIGHT_F_ENABLE specular color's f enable/disable

- **0** for D3DM
- **1** for OpenGL-ES

M3D_LIGHT_INF_POS To determine the infinite view vector.

- **0** Set infinite view vector [0 0 1]
- **1** Set infinite view vector [0 0 -1]

M3D_BASE+0F8h M3D texture image #0 height/width/type/format register M3D_TEX_IMG_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE						M3D_TEX_IMG_HEIGHT_0									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_WIDTH_0										RESERVE	M3D_TEX_BYTE_ORDER_0	M3D_TEX_IMG_FORMAT_0			
Type	R/W											R/W	R/W			

M3D_TEX_IMG_FORMAT_0 Specifies the format of texture image #0

0	R8G8B8
1	R5G6B5
2	A8R8G8B8
3	X8R8G8B8
4	A1R5G5B5
5	X1R5G5B5
6	L8
7	A8
8	A4R4G4B4
9	X4R5G4B4
10	L8A8
11	(reserved)
12	(reserved)
13	DXT1

14	DXT2
15	DXT3
M3D_TEX_BYTE_ORDER_0	Specifies the byte order of texture image #0
0	OpenGL ES
1	D3DM
M3D_TEX_IMG_WIDTH_0	Specifies the width of texture image #0
M3D_TEX_IMG_HEIGHT_0	Specifies the height of texture image #0

M3D_BASE+0FCh M3D texture image #1 height/width/type/format register M3D_TEX_IMG_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE						M3D_TEX_IMG_HEIGHT_1									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_WIDTH_1										RESERVE	M3D_TEX_BYTE_ORDER_1	M3D_TEX_IMG_FORMAT_1			
Type	R/W											R/W	R/W			

M3D_TEX_IMG_FORMAT_1	Specifies the format of texture image #1
M3D_TEX_BYTE_ORDER_1	Specifies the byte order of texture image #1
M3D_TEX_IMG_WIDTH_1	Specifies the width of texture image #1
M3D_TEX_IMG_HEIGHT_1	Specifies the height of texture image #1

M3D_BASE+100h M3D texture image #2 height/width/type/format register M3D_TEX_IMG_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE						M3D_TEX_IMG_HEIGHT_2									
Type							R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_WIDTH_2										RESERVE	M3D_TEX_BYTE_ORDER_2	M3D_TEX_IMG_FORMAT_2			
Type	R/W												R/W			

M3D_TEX_IMG_FORMAT_2	Specifies the format of texture image #2
M3D_TEX_BYTE_ORDER_2	Specifies the byte order of texture image #2
M3D_TEX_IMG_WIDTH_2	Specifies the width of texture image #2
M3D_TEX_IMG_HEIGHT_2	Specifies the height of texture image #2

M3D_BASE+104h M3D texture control register M3D_TEX_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						M3D_TEX_LOD_HALF_EN_2	M3D_TEX_LOD_HALF_EN_1	M3D_TEX_LOD_HALF_EN_0	M3D_TEX_THRESHOLD_2	M3D_TEX_THRESHOLD_EN_1	M3D_TEX_THRESHOLD_EN_0	M3D_MAX_LOD_2				M3D_MAX_LOD_1

Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W
Reset						1	1	1	1	1	1	0				0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_MAX_LOD_1**				**M3D_MAX_LOD_0**			**M3D_SPECULAR_ENABLE**	**M3D_TEX_STAGE_EN**				**M3D_TEX_ENABLE**			
Type	R/W				R/W			R/W	R/W				R/W			
Reset	0				0			0	0				0			

M3D_TEX_ENABLE Texture disable/enable. (0 represent disable, 1 represent enable)

M3D_TEX_STAGE_EN Texture stage disable/enable. (0 represent disable, 1 represent enable)

M3D_SPECULAR_ENABLE D3DM_SPECULARENABLE

M3D_MAX_LOD_0 D3DMTSS_MAXMIPLEVEL of texture #0

M3D_MAX_LOD_1 D3DMTSS_MAXMIPLEVEL of texture #1

M3D_MAX_LOD_2 D3DMTSS_MAXMIPLEVEL of texture #2

M3D_TEX_THRESHOLD_EN_0 Magnification threshold 0.5 of texture #0

M3D_TEX_THRESHOLD_EN_1 Magnification threshold 0.5 of texture #1

M3D_TEX_THRESHOLD_EN_2 Magnification threshold 0.5 of texture #2

M3D_TEX_LOD_HALF_EN_0 Minification less than 0.5 of texture #0

M3D_TEX_LOD_HALF_EN_1 Minification less than 0.5 of texture #1

M3D_TEX_LOD_HALF_EN_2 Minification less than 0.5 of texture #2

M3D_BASE+108h M3D texture image #0 level #0 address M3D_TEX_IMG_PTR_0_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_IMG_PTR_0_0[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_IMG_PTR_0_0[15:0]**															
Type	R/W															

M3D_TEX_IMG_PTR_0_0 Specifies the address of texture image #0 level #0

M3D_BASE+10Ch M3D texture image #0 level #1 address M3D_TEX_IMG_PTR_0_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_IMG_PTR_0_1[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_IMG_PTR_0_1[15:0]**															
Type	R/W															

M3D_TEX_IMG_PTR_0_1 Specifies the address of texture image #0 level #1

M3D_BASE+110h M3D texture image #0 level #2 address M3D_TEX_IMG_PTR_0_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_2[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_2 Specifies the address of texture image #0 level #2

M3D_BASE+114h M3D texture image #0 level #3 address M3D_TEX_IMG_PTR_0_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_3[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_3 Specifies the address of texture image #0 level #3

M3D_BASE+118h M3D texture image #0 level #4 address M3D_TEX_IMG_PTR_0_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_4[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_4 Specifies the address of texture image #0 level #4

M3D_BASE+11Ch M3D texture image #0 level #5 address M3D_TEX_IMG_PTR_0_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_5[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_5 Specifies the address of texture image #0 level #5

M3D_BASE+120h M3D texture image #0 level #6 address M3D_TEX_IMG_PTR_0_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_6[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_6[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_6 Specifies the address of texture image #0 level #6

M3D_BASE+124h M3D texture image #0 level #7 address M3D_TEX_IMG_PTR_0_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_7[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_7 Specifies the address of texture image #0 level #7

M3D_BASE+128h M3D texture image #0 level #8 address M3D_TEX_IMG_PTR_0_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_0_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_0_8[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_0_8 Specifies the address of texture image #0 level #8

M3D_BASE+12Ch M3D texture image #1 level #0 address M3D_TEX_IMG_PTR_1_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_0[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_0 Specifies the address of texture image #1 level #0

M3D_BASE+130h M3D texture image #1 level #1 address M3D_TEX_IMG_PTR_1_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_1[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_1 Specifies the address of texture image #1 level #1

© 2009 MediaTek Inc.

M3D_BASE+134h M3D texture image #1 level #2 address M3D_TEX_IMG_PTR_1_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_2[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_2 Specifies the address of texture image #1 level #2

M3D_BASE+138h M3D texture image #1 level #3 address M3D_TEX_IMG_PTR_1_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_3[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_3 Specifies the address of texture image #1 level #3

M3D_BASE+13Ch M3D texture image #1 level #4 address M3D_TEX_IMG_PTR_1_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_4[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_4 Specifies the address of texture image #1 level #4

M3D_BASE+140h M3D texture image #1 level #5 address M3D_TEX_IMG_PTR_1_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_5[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_5 Specifies the address of texture image #1 level #5

M3D_BASE+144h M3D texture image #1 level #6 address M3D_TEX_IMG_PTR_1_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_6[31:16]															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_6[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_6 Specifies the address of texture image #1 level #6

M3D_BASE+148h M3D texture image #1 level #7 address M3D_TEX_IMG_PTR_1_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_7[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_7 Specifies the address of texture image #1 level #7

M3D_BASE+14Ch M3D texture image #1 level #8 address M3D_TEX_IMG_PTR_1_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_1_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_1_8[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_1_8 Specifies the address of texture image #1 level #8

M3D_BASE+150h M3D texture image #2 level #0 address M3D_TEX_IMG_PTR_2_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_0[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_0 Specifies the address of texture image #2 level #0

M3D_BASE+154h M3D texture image #2 level #1 address M3D_TEX_IMG_PTR_2_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_1[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_1 Specifies the address of texture image #2 level #1

M3D_BASE+158h M3D texture image #2 level #2 address M3D_TEX_IMG_PTR_2_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_2[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_2 Specifies the address of texture image #2 level #2

M3D_BASE+15Ch M3D texture image #2 level #3 address M3D_TEX_IMG_PTR_2_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_3[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_3 Specifies the address of texture image #2 level #3

M3D_BASE+160h M3D texture image #2 level #4 address M3D_TEX_IMG_PTR_2_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_4[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_4 Specifies the address of texture image #2 level #4

M3D_BASE+164h M3D texture image #2 level #5 address M3D_TEX_IMG_PTR_2_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_5[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_5 Specifies the address of texture image #2 level #5

M3D_BASE+168h M3D texture image #2 level #6 address M3D_TEX_IMG_PTR_2_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_6[31:16]															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_6[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_6 Specifies the address of texture image #2 level #6

M3D_BASE+16Ch M3D texture image #2 level #7 address M3D_TEX_IMG_PTR_2_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_7[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_7 Specifies the address of texture image #2 level #7

M3D_BASE+170h M3D texture image #2 level #8 address M3D_TEX_IMG_PTR_2_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_IMG_PTR_2_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_IMG_PTR_2_8[15:0]															
Type	R/W															

M3D_TEX_IMG_PTR_2_8 Specifies the address of texture image #2 level #8

M3D_BASE+174h\|974h M3D color r register M3D_COLOR_R_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_R_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_R_0[15:0]															
Type	R/W															

M3D_COLOR_R_0 Set the current color R. (0088h : data format is s15.16, 2088h : data format is s[8].23)

M3D_BASE+178h\|978h M3D color g register M3D_COLOR_G_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_G_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_G_0[15:0]															
Type	R/W															

M3D_COLOR_G_0 Set the current color G. (008Ch : data format is s15.16, 208Ch : data format is s[8].23)

M3D_BASE+17Ch|97Ch M3D color b register M3D_COLOR_B_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_B_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_B_0[15:0]															
Type	R/W															

M3D_COLOR_B_0 Set the current color B. (0090h : data format is s15.16, 2090h : data format is s[8].23)

M3D_BASE+180h|980h M3D color a register M3D_COLOR_A_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_A_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_A_0[15:0]															
Type	R/W															

M3D_COLOR_A_0 Set the current color A. (0094h : data format is s15.16, 2094h : data format is s[8].23)

M3D_BASE+184h|984h M3D color r register M3D_COLOR_R_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_R_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_R_1[15:0]															
Type	R/W															

M3D_COLOR_R_1 Set the current color R. (0088h : data format is s15.16, 2088h : data format is s[8].23)

M3D_BASE+188h|988h M3D color g register M3D_COLOR_G_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_G_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_G_1[15:0]															
Type	R/W															

M3D_COLOR_G_1 Set the current color G. (008Ch : data format is s15.16, 208Ch : data format is s[8].23)

M3D_BASE+18Ch|98Ch M3D color b register M3D_COLOR_B_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_B_1[31:16]															

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_B_1[15:0]															
Type	R/W															

M3D_COLOR_B_1 Set the current color B. (0090h : data format is s15.16, 2090h : data format is s[8].23)

M3D_BASE+190h|990h M3D color a register M3D_COLOR_A_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_COLOR_A_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_COLOR_A_1[15:0]															
Type	R/W															

M3D_COLOR_A_1 Set the current color A. (0094h : data format is s15.16, 2094h : data format is s[8].23)

M3D_BASE+198h M3D scissor test enable/disable Register M3D_SCISSOR_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																M3D_SCISSOR_ENABLE
Type																R/W
Reset																0

M3D_SCISSOR_ENABLE Scissor test disable/enable. (0 represent disable, 1 represent enable)

M3D_BASE+19Ch M3D scissor test left boundary Register M3D_SCISSOR_LEFT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							M3D_SCISSOR_LEFT[9:0]									
Type							R/W									
Reset							0									

M3D_SCISSOR_LEFT Scissor box left boundary

M3D_BASE+1A0h M3D scissor test bottom boundary Register M3D_SCISSOR_BOTTOM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							M3D_SCISSOR_BOTTOM[9:0]									
Type							R/W									
Reset							0									

M3D_SCISSOR_BOTTOM Scissor box bottom boundary

M3D_BASE+1A4h M3D scissor test right boundary Register M3D_SCISSOR_RIGHT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							M3D_SCISSOR_RIGHT[9:0]									
Type							R/W									
Reset							0									

M3D_SCISSOR_RIGHT Scissor box right boundary

M3D_BASE+1A8h M3D scissor test top boundary Register M3D_SCISSOR_TOP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							M3D_SCISSOR_TOP[9:0]									
Type							R/W									
Reset							0									

M3D_SCISSOR_TOP Scissor box top boundary

M3D_BASE+1ACh M3D alpha test register M3D_ALPHA_TEST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					M3D_ALPHA_TEST_REF[7:0]								M3D_ALPHA_FUNC[2:0]			M3D_ALPHA_TEST_ENABLE
Type					R/W								R/W			R/W
Reset					0								0			0

M3D_ALPHA_TEST_ENABLE Enable alpha test.
M3D_ALPHA_TEST_FUNC The alpha test function.

000 NEVER
001 LESS
010 EQUAL
011 LEQUAL
100 GREATER
101 NOTEQUAL
110 GEQUAL
111 ALWAYS

M3D_ALPHA_TEST_REF The alpha test reference value.

M3D_BASE+1B0h M3D stencil test register M3D_STENCIL_TEST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			M3D_STENCIL_TEST_FMT	M3D_STENCIL_TEST_DPPASS_OP			M3D_STENCIL_TEST_DPFAIL_OP			M3D_STENCIL_TEST_SFAIL_OP			M3D_STENCIL_TEST_MASK[7:4]			
Type			R/W	R/W			R/W			R/W			R/W			
Reset			0	0			0			0			0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_STENCIL_TEST_MASK[3:0]				M3D_STENCIL_TEST_REF[7:0]								M3D_STENCIL_TEST_FUNC[2:0]			M3D_STENCILTEST_ENABLE
Type	R/W				R/W								R/W			R/W
Reset	0				0								0			0

M3D_STENCIL_TEST_ENABLE Enable stencil test.

M3D_STENCIL_TEST_FUNC Stencil test function

000 NEVER
001 LESS
010 EQUAL
011 LEQUAL
100 GREATER
101 NOTEQUAL
110 GEQUAL
111 ALWAYS

M3D_STENCIL_TEST_REF Stencil test reference value.

M3D_STENCIL_TEST_MASK Stencil test mask.

M3D_STENCIL_TEST_SFAIL_OP The stencil operation when stencil test fails.

000 KEEP
001 REPLACE
010 INCR
011 DECR
100 ZERO

101 INVERT
110 INCR_WRAP
111 DECR_WRAP

M3D_STENCIL_TEST_DPFAIL_OP The stencil operation when depth test fails.

000 KEEP
001 REPLACE
010 INCR
011 DECR
100 ZERO
101 INVERT
110 INCR_WRAP
111 DECR_WRAP

M3D_STENCIL_TEST_DPPASS_OP The stencil operation when depth test passes.

000 KEEP
001 REPLACE
010 INCR
011 DECR
100 ZERO
101 INVERT
110 INCR_WRAP
111 DECR_WRAP

M3D_STENCIL_TEST_FMT The stencil value format.

0 8-bits
1 1-bit.

M3D_BASE+1B4h M3D depth test register M3D_DEPTH_TEST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												M3D_Z_MASK	M3D_DEPTH_FUNC[2:0]			M3D_DEPTHA_TEST_ENABLE
Type												R/W	R/W			R/W
Reset												0	0			0

M3D_DEPTH_TEST_ENABLE Enable the depth test.

M3D_DEPTH_TEST_FUNC Select the depth function

000 NEVER
001 LESS
010 EQUAL

011	LEQUAL
100	GREATER
101	NOTEQUAL
110	GEQUAL
111	ALWAYS

M3D_Z_TEST_MASK Z buffer mask. Set this bit to 0 to disable z buffer output.

M3D_BASE+1B8h M3D blending register M3D_BLEND

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	M3D_BLEND_OP			M3D_R_MASK	M3D_G_MASK	M3D_B_MASK	M3D_A_MASK	M3D_BLEND_FUNC_DEST[3:0]				M3D_BLEND_FUNC_SRC[3:0]				M3D_BLEND_ENABLE
Type	R/W			R/W	R/W	R/W	R/W					R/W				R/W
Reset	0			1	1	1	1					0				0

M3D_BLEND_ENABLE Enable the blending function.

M3D_BLEND_FUNC_SRC Select the blending function for source color.

0000	BLEND_SRC_COLOR
0001	BLEND_ONE_MINUS_SRC_COLOR
0010	BLEND_SRC_ALPHA
0011	BLEND_ONE_MINUS_SRC_ALPHA
0100	BLEND_DST_ALPHA
0101	BLEND_ONE_MINUS_DST_ALPHA
0110	BLEND_DST_COLOR
0111	BLEND_ONE_MINUS_DST_COLOR
1000	BLEND_SRC_ALPHA_SATURATE
1001	BLEND_ZERO
1010	BLEND_ONE

M3D_BLEND_FUNC_DEST Select the blending function for destination color.

0000	BLEND_SRC_COLOR
0001	BLEND_ONE_MINUS_SRC_COLOR
0010	BLEND_SRC_ALPHA
0011	BLEND_ONE_MINUS_SRC_ALPHA
0100	BLEND_DST_ALPHA
0101	BLEND_ONE_MINUS_DST_ALPHA
0110	BLEND_DST_COLOR
0111	BLEND_ONE_MINUS_DST_COLOR
1000	BLEND_SRC_ALPHA_SATURATE
1001	BLEND_ZERO
1010	BLEND_ONE

M3D_A_MASK The alpha color mask. Set this bit to 0 will disable the alpha channel output to color buffer.
M3D_B_MASK The blue color mask. Set this bit to 0 will disable the blue channel output to color buffer.
M3D_G_MASK The green color mask. Set this bit to 0 will disable the green channel output to color buffer.
M3D_R_MASK The red color mask. Set this bit to 0 will disable the red channel output to color buffer.
M3D_BLEND_OP Select the blending operation

- **000** C = Cs*S + Cd*D
- **001** C = Cs*S–Cd*D
- **010** C = Cd*D–Cs*S
- **011** C = min(Cs,Cd)
- **100** C = max(Cs,Cd)

M3D_BASE+1BCh M3D logic operation register M3D_LOGIC_OP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												M3D_LOGIC_OP_FUNC[3:0]				M3D_LOGIC_OP_ENABLE
Type												R/W	R/W			R/W
Reset												0	0			0

M3D_LOGIC_OP_ENABLE Enable the logic operation.
M3D_LOGIC_OP_FUNC Select the logic operation function

- **0000** CLEAR
- **0001** AND
- **0010** AND_REVERSE
- **0011** COPY
- **0100** AND_INVERTED
- **0101** NOOP
- **0110** XOR
- **0111** OR
- **1000** NOR
- **1001** EQUIV
- **1010** INVERT
- **1011** OR_REVERSE
- **1100** COPY_INVERTED
- **1101** OR_INVERTED
- **1110** NAND

 © 2009 MediaTek Inc.

1111 SET

M3D_BASE+1C0h M3D frame buffer format register M3D_FRAME_BUF_FORMAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name														M3D_FRAME_BUF_FORMAT		
Type														R/W		
Reset														0		

M3D_FRAME_BUF_FORMAT Select the format of color buffer

- **001** RGB_888
- **010** RGB565
- **100** ARGB8888

M3D_BASE+1C4h M3D frame buffer address register M3D_FRAME_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_FRAME_BUF_ADDR[31:16]															
Type	R/W															
	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	M3D_FRAME_BUF_ADDR[15:0]															
Type	R/W															
Reset	0															

M3D_FRAME_BUF_ADDR Color buffer address.

M3D_BASE+1C8h M3D frame buffer width register M3D_FRAME_BUF_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							M3D_FRAME_BUF_WIDITH[9:0]									
Type							R/W									
Reset							0									

M3D_FRAME_BUF_WIDTH Frame buffer width.

M3D_BASE+1CCh M3D frame buffer height register M3D_FRAME_BUF_HEIGHT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							M3D_FRAME_BUF_HEIGHT[9:0]									
Type							R/W									
Reset							0									

M3D_FRAME_BUF_HEIGHT Frame buffer height.

M3D_BASE+1D0h M3D stencil buffer write mask Register M3D_STENCIL_WT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									M3D_STENCIL_WT_MASK							
Type									R/W							
Reset									FF							

M3D_STENCIL_WT_MASK Stencil buffer write mask after stencil operation.

M3D_BASE+1D4h M3D depth buffer address register M3D_DEPTH_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DEPTH_BUF_ADDR[31:16]															
Type	R/W															
	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DEPTH_BUF_ADDR[15:0]															
Type	R/W															
Reset	0															

M3D_DEPTH_BUF_ADDR Depth buffer address.

M3D_BASE+1D8h M3D stencil buffer address register M3D_STENCIL_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_STENCIL_BUF_ADDR[31:16]															
Type	R/W															
	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_STENCIL_BUF_ADDR[15:0]															
Type	R/W															
Reset	0															

M3D_STENCIL_BUF_ADDR Stencil buffer address.

M3D_BASE+1DCh M3D depth buffer clear value register M3D_DEPTH_CLEAR_VAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DEPTH_CLEAR_VAL[15:0]															
Type	R/W															
Reset	0															

M3D_DEPTH_CLEAR_VAL Depth buffer clear value.

M3D_BASE+1E8h M3D earlyz test enable/disable Register M3D_EARLY_Z_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															M3D_EARLYZ_MODE	M3D_EARLY_Z_ENABLE
Type															R/W	R/W
Reset															0	0

M3D_EARLY_Z_ENABLE EarlyZ test disable/enable. (0 represent disable, 1 represent enable)
EarlyZ must be disable when alpha test/stencil test/line/point is enable
Once EarlyZ is disable at some list, it must be disable until the frame buffer is flush
M3D_EARLYZ_MODE EarlyZ mode. (0 specifies each earlyz buffer unit contains 4 pix, 1 specifies each earlyz buffer unit contains 8 pix)
Set m3d_earlyz_mode=0, better performance but more SYSRAM usage
Set m3d_earlyz_mode=1, worse performance but less SYSRAM usage
If SYSRAM is big enough, set m3d_earlyz_mode 0 is recommended

M3D_BASE+1ECh M3D earlyz buffer address Register M3D_EARLY_Z_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_EARLY_Z_BUF_ADDR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_EARLY_Z_BUF_ADDR[15:0]															
Type	R/W															

M3D_EARLY_Z_BUF_ADDR Specifies the memory address of the EarlyZ buffer
For VGA size, EarlyZ buffer is 76.8KB(EarlyZ mode 0) and 38.4KB(EarlyZ mode 1)
For QVGA size, EarlyZ buffer is 38.4KB(EarlyZ mode 0) and 19.2KB(EarlyZ mode 1)
If EarlyZ is enable, clear depth buffer and EarlyZ buffer at the same time. The clear value of EarlyZ buffer is {1'b0,M3D_DEPTH_CLEAR_VAL[15:9]}

M3D_BASE+1FCh M3D INTSTA_WACK M3D_INTSTA_WTACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED									M3D_COLOR_FLUSH_DONE_INTSTA_WTACK	M3D_Z_FLUSH_DONE_INTSTA_WTACK	RESERVED	RESERVED	RESERVED	RESERVED	M3D_RENDER_DONE_INTSTA_WTACK
Type										W	W					W

M3D_RENDER_DONE_INTSTA MCU Clear Interrupting status of rendering.
M3D_Z_FLUSH_DONE_INTSTA MCU Clear Interrupting status of z cache flush.
M3D_COLOR_FLUSH_DONE_INTSTA MCU Clear Interrupting status of color cache flush.

M3D_BASE+200h|A00h M3D texture image #0 environment RED color M3D_TEX_ENV_COLOR_R_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_R_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_R_0[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_R_0 Specifies the environment RED color of texture image #0

M3D_BASE+204h|A04h M3D texture image #0 environment GREEN color M3D_TEX_ENV_COLOR_G_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_G_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_G_0[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_G_0 Specifies the environment GREEN color of texture image #0

M3D_BASE+208h|A08h M3D texture image #0 environment BLUE color M3D_TEX_ENV_COLOR_B_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_B_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_B_0[15:0]															
Type	R/W															

© 2009 MediaTek Inc.

M3D_TEX_ENV_COLOR_B_0 Specifies the environment BLUE color of texture image #0

M3D_BASE+20Ch|A0Ch M3D texture image #0 environment ALPHA color M3D_TEX_ENV_COLOR_A_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_A_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_A_0[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_A_0 Specifies the environment ALPHA color of texture image #0

M3D_BASE+210h|A10h M3D texture image #1 environment RED color M3D_TEX_ENV_COLOR_R_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_R_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_R_1[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_R_1 Specifies the environment RED color of texture image #1

M3D_BASE+214h|A14h M3D texture image #1 environment GREEN color M3D_TEX_ENV_COLOR_G_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_G_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_G_1[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_G_1 Specifies the environment GREEN color of texture image #1

M3D_BASE+218h|A18h M3D texture image #1 environment BLUE color M3D_TEX_ENV_COLOR_B_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_B_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_B_1[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_B_1 Specifies the environment BLUE color of texture image #1

M3D_BASE+21Ch|A1Ch M3D texture image #1 environment ALPHA color M3D_TEX_ENV_COLOR_A_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_A_1[31:16]															

© 2009 MediaTek Inc.

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_A_1[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_A_1 Specifies the environment ALPHA color of texture image #1

M3D_BASE+220h|A20h M3D texture image #2 environment RED color M3D_TEX_ENV_COLOR_R_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_R_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_R_2[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_R_2 Specifies the environment RED color of texture image #2

M3D_BASE+224h|A24h M3D texture image #2 environment GREEN color M3D_TEX_ENV_COLOR_G_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_G_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_G_2[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_G_2 Specifies the environment GREEN color of texture image #2

M3D_BASE+228h|A28h M3D texture image #2 environment BLUE color M3D_TEX_ENV_COLOR_B_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_B_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_B_2[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_B_2 Specifies the environment BLUE color of texture image #2

M3D_BASE+22Ch|A2Ch M3D texture image #2 environment ALPHA color M3D_TEX_ENV_COLOR_A_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_ENV_COLOR_A_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_ENV_COLOR_A_2[15:0]															
Type	R/W															

M3D_TEX_ENV_COLOR_A_2 Specifies the environment ALPHA color of texture image #2

M3D_BASE+230h|A30h M3D texture #0 mipmap LOD bias M3D_LOD_BIAS_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_LOD_BIAS_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_LOD_BIAS_0[15:0]															
Type	R/W															

M3D_LOD_BIAS_0 Specifies the mipmap LOD bias of texture image #0

M3D_BASE+234h|A34h M3D texture #1 mipmap LOD bias M3D_LOD_BIAS_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_LOD_BIAS_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_LOD_BIAS_1[15:0]															
Type	R/W															

M3D_LOD_BIAS_1 Specifies the mipmap LOD bias of texture image #1

M3D_BASE+238h|A38h M3D texture #2 mipmap LOD bias M3D_LOD_BIAS_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_LOD_BIAS_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_LOD_BIAS_2[15:0]															
Type	R/W															

M3D_LOD_BIAS_2 Specifies the mipmap LOD bias of texture image #2

M3D_BASE+23Ch M3D vertex cache hit counter M3D_VC_HIT_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VC_HIT_CNT[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VC_HIT_CNT[15:0]															
Type	R/W															

M3D_VC_HIT_CNT Vertex cache hit counter.

M3D_BASE+240h M3D texture image #0 scale and texture operation M3D_TEX_OP_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RESERVE														M3D_TEX_OP_DST_A_0
Type															R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_OP_DST_RGB_0	M3D_TEX_DOT3_RGBA_0	M3D_TEX_A_SCALE_0		M3D_TEX_RGB_SCALE_0		M3D_TEX_OP_A_0					M3D_TEX_OP_RGB_0				
Type	R/W	R/W	R/W		R/W		R/W					R/W				

M3D_TEX_OP_RGB_0 Specifies RGB operation of texture image #0

0	SELECTARG1
1	SELECTARG2
2	MODULATE
3	ADD
4	ADDSIGNED
5	SUBTRACT
6	PREMODULATE
7	DOTPRODUCT3
8	MULTIPLYADD
9	LERP

M3D_TEX_OP_A_0 Specifies A operation of texture image #0

M3D_TEX_ RGB_SCALE_0 Specifies RGB scale of texture image #0

M3D_TEX_ A_SCALE_0 Specifies A scale of texture image #0

M3D_TEX_ DOT3_RGBA_0 Specifies DOT3_RGBA environment mode of texture image #0

M3D_TEX_OP_DST_RGB_0 Specifies RGB output destination of texture image #0

0	DEFAULT (CURRENT)
1	TEMP

M3D_TEX_OP_DST_A_0 Specifies A output destination of texture image #0

OpenGL ES Texture Functions:

// initialization

M3D_TEX_OP_RGB_0 =

M3D_TEX_OP_A_= M3D_TEX_OP_SELECTARG1;

M3D_TEX_SRC0_RGB_0 =

M3D_TEX_SRC1_RGB_0 =

M3D_TEX_SRC2_RGB_0 =

M3D_TEX_SRC0_A_0 =

M3D_TEX_SRC1_A_0 =

M3D_TEX_SRC2_A_0 = M3D_TEX_SRC_PRIMARY;

M3D_TEX_OPD0_RGB_0 =

M3D_TEX_OPD1_RGB_1 =

M3D_TEX_OPD2_RGB_2 = M3D_TEX_OPD_SRC_COLOR;

M3D_TEX_OPD0_A_0 =

```
M3D_TEX_OPD1_A_0 =
M3D_TEX_OPD2_A_0 = M3D_TEX_OPD_SRC_ALPHA;
M3D_TEX_RGB_SCALE_0 = 0;
M3D_TEX_A_SCALE_0 = 0;
M3D_TEX_DOT3_RGBA_0 = FALSE;
switch (texture function)
{
case REPLACE:
  // RGB part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_TEXTURE;
      break;
    default:
      assert(false);
      break;
    }
  }
  // A part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
          M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_OP_SRC1_A_0 = M3D_TEX_SRC_TEXTURE;
      break;
```

```
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
case MODULATE:
  // RGB part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_MODULATE;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      M3D_TEX_SRC2_RGB_0 = M3D_TEX_SRC_TEXTURE;
      break;
    default:
      assert(false);
      break;
    }
  }
  // A part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_A8R8G8B8:
```

```
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_MODULATE;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      M3D_TEX_SRC2_A_0 = M3D_TEX_SRC_TEXTURE;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
  break;
case DECAL:
   // RGB part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_TEXTURE;
      break;
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_LERP;
      M3D_TEX_SRC0_RGB_0 = M3D_TEX_SRC_TEXTURE;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_TEXTURE;
      M3D_TEX_SRC2_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      M3D_TEX_OPD0_RGB_0 = M3D_TEX_OPD_SRC_ALPHA;
      break;
    default:
      assert(false);
      break;
    }
  }
  // A part
```

```
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
  break;
case BLEND:
  // RGB part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_LERP;
      M3D_TEX_SRC0_RGB_0 = M3D_TEX_SRC_TEXTURE;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_CONSTANT;
      M3D_TEX_SRC2_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
```

```
  // A part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_MODULATE;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      M3D_TEX_SRC2_A_0 = M3D_TEX_SRC_TEXTURE:
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC1_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
  break;
case ADD:
  // RGB part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B5:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_ADD;
      M3D_TEX_SRC1_RGB_0 = M3D_TEX_SRC1_PREVIOUS;
```

```
      M3D_TEX_SRC2_RGB_0 = M3D_TEX_SRC2_TEXTURE;
      break;
    default:
      assert(false);
      break;
    }
  }
  // A part
  {
    switch (M3D_TEX_IMG_0[3:0])
    {
    case M3D_TEX_FMT_A8:
    case M3D_TEX_FMT_L8A8:
    case M3D_TEX_FMT_A8R8G8B8:
    case M3D_TEX_FMT_A1R5G5B5:
    case M3D_TEX_FMT_A4R4G4B4:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_MODULATE;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC1_PREVIOUS;
      M3D_TEX_SRC2_A_0 = M3D_TEX_SRC2_TEXTURE;
      break;
    case M3D_TEX_FMT_L8:
    case M3D_TEX_FMT_R8G8B8:
    case M3D_TEX_FMT_R5G6B8:
      M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_A_0 = M3D_TEX_SRC_PREVIOUS;
      break;
    default:
      assert(false);
      break;
    }
  }
  break;
case COMBINE:
  // RGB part
  {
    switch (COMBINE_RGB)
    {
    case REPLACE:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SELECTARG1;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      break;
    case MODULATE:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_MODULATE;
```

```
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case ADD:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_ADD;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case ADD_SIGNED:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_ADDSIGNED;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case INTERPOLATE:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_LERP;
      M3D_TEX_SRC0_RGB_0 = SRC2_RGB_0
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case SUBTRACT:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_SUBTRACT;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case DOT3_RGB:
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_DOTPRODUCT3;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    case DOT3_RGBA:
      M3D_TEX_DOT3_RGBA = TRUE;
      M3D_TEX_OP_RGB_0 = M3D_TEX_OP_DOTPRODUCT3;
      M3D_TEX_SRC1_RGB_0 = SRC0_RGB_0;
      M3D_TEX_SRC2_RGB_0 = SRC1_RGB_0;
      break;
    default:
      assert(false);
      break;
    }
  }
  // A part
  {
    switch (COMBINE_A)
```

```
  {
  case REPLACE:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_SELECTARG1;
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    break;
  case MODULATE:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_MODULATE;
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    M3D_TEX_SRC2_A_0 = SRC1_A_0;
    break;
  case ADD:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_ADD;
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    M3D_TEX_SRC2_A_0 = SRC1_A_0;
    break;
  case ADD_SIGNED:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_ADDSIGNED;
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    M3D_TEX_SRC2_A_0 = SRC1_A_0;
    break;
  case INTERPOLATE:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_LERP;
    M3D_TEX_SRC0_A_0 = SRC2_A_0
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    M3D_TEX_SRC2_A_0 = SRC1_A_0;
    break;
  case SUBTRACT:
    M3D_TEX_OP_A_0 = M3D_TEX_OP_SUBTRACT;
    M3D_TEX_SRC1_A_0 = SRC0_A_0;
    M3D_TEX_SRC2_A_0 = SRC1_A_0;
    break;
  default:
    assert(false);
    break;
  }
}
M3D_TEX_OPD0_RGB_0 = OPERAND2_RGB_0;
M3D_TEX_OPD1_RGB_0 = OPERAND0_RGB_0;
M3D_TEX_OPD2_RGB_0 = OPERAND1_RGB_0;
M3D_TEX_OPD0_A_0 = OPERAND2_A_0;
M3D_TEX_OPD1_A_0 = OPERAND0_A_0;
M3D_TEX_OPD2_A_0 = OPERAND1_A_0;
M3D_TEX_RGB_SCALE_0 = SCALE_RGB_0;
```

```
  M3D_TEX_A_SCALE_0 = SCALE_A_0;
  break;
default:
  assert(false);
  break;
}
```

M3D_BASE+244h M3D texture image #1 scale and texture operation M3D_TEX_OP_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															M3D_TEX_OP_DST_A_1
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_OP_DST_RGB_1	M3D_TES_DOT3_RGBA_1	M3D_TEX_A_SCALE_1		M3D_TEX_RGB_SCALE_1		M3D_TEX_OP_A_1					M3D_TEX_OP_RGB_1				
Type	R/W	R/W	R/W		R/W		R/W					R/W				

M3D_TEX_OP_RGB_1 Specifies RGB operation of texture image #1
M3D_TEX_OP_A_1 Specifies A operation of texture image #1
M3D_TEX_ RGB_SCALE_1 Specifies RGB scale of texture image #1
M3D_TEX_ A_SCALE_1 Specifies A scale of texture image #1
M3D_TEX_ DOT3_RGBA_1 Specifies DOT3_RGBA environment mode of texture image #1
M3D_TEX_ OP_DST_RGB_1 Specifies RGB output destination of texture image #1
M3D_TEX_ OP_DST_A_1 Specifies A output destination of texture image #1

M3D_BASE+248h M3D texture image #2 scale and texture operation M3D_TEX_OP_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															M3D_TEX_OP_DST_A_2
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_OP_DST_RGB_2	M3D_TEX_DOT3_RGBA_2	M3D_TEX_A_SCALE_2		M3D_TEX_RGB_SCALE_2		M3D_TEX_OP_A_2					M3D_TEX_OP_RGB_2				
Type	R/W	R/W	R/W		R/W		R/W					R/W				

M3D_TEX_OP_RGB_2 Specifies the texture function of texture image #2
M3D_TEX_OP_A_2 Specifies A combine function of texture image #2

M3D_TEX_ RGB_SCALE_2 Specifies RGB scale of texture image #2
M3D_TEX_ A_SCALE_2 Specifies A scale of texture image #2
M3D_TEX_ DOT3_RGBA_2 Specifies DOT3_RGBA environment mode of texture image #2
M3D_TEX_ OP_DST_RGB_2 Specifies RGB output destination of texture image #2
M3D_TEX_ OP_DST_A_2 Specifies A output destination of texture image #2

M3D_BASE+24Ch M3D vertex cache vertex counter M3D_VC_VTX_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VC_VTX_CNT[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VC_VTX_CNT[15:0]															
Type	R/W															

M3D_VC_VTX_CNT Vertex cache vertex counter.

M3D_BASE+250h M3D texture image #0 RGBA source M3D_TEX_SRC_OPD_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE		M3D_TEX_OPD_RGBA_0												M3D_TEX_SRC_RGBA_0[17:16]	
Type			R/W												R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_SRC_RGBA_0[15:0]															
Type	R/W															

M3D_TEX_SRC_RGBA_0 Specifies source of texture image #0
M3D_TEX_SRC0_RGB[2:0] Specifies RGB source #0 of texture image #0
M3D_TEX_SRC1_RGB[5:3] Specifies RGB source #1 of texture image #0
M3D_TEX_SRC2_RGB[8:6] Specifies RGB source #2 of texture image #0
M3D_TEX_SRC0_A[11:9] Specifies A source #0 of texture image #0
M3D_TEX_SRC1_A[14:12] Specifies A source #1 of texture image #0
M3D_TEX_SRC2_A[17:15] Specifies A source #2 of texture image #0
000 M3D_TEX_SRC_TEXTURE
001 M3D_TEX_SRC_CONSTANT
010 M3D_TEX_SRC_PRIMARY
011 M3D_TEX_SRC_PREVIOUS (Current)
100 M3D_TEX_SRC_SECONDARY
101 M3D_TEX_SRC_TEMP
110 M3D_TEX_SRC_ZERO
M3D_TEX_OPD_RGBA_0 Specifies operand of texture image #0
M3D_TEX_OPD0_RGB[1:0] Specifies RGB operand #0 of texture image #0
M3D_TEX_OPD1_RGB[3:2] Specifies RGB operand #1 of texture image #0
M3D_TEX_OPD2_RGB[5:4] Specifies RGB operand #2 of texture image #0
M3D_TEX_OPD0_A[7:6] Specifies A operand #0 of texture image #0

M3D_TEX_OPD1_A[9:8] Specifies A operand #1 of texture image #0
M3D_TEX_OPD2_A[11:10] Specifies A operand #2 of texture image #0

00	M3D_TEX_OPD_SRC_COLOR
01	M3D_TEX_ONE_MINUS_SRC_COLOR
10	M3D_TEX_SRC_ALPHA
11	M3D_TEX_ONE_MINUS_SRC_ALPHA

M3D_BASE+254h M3D texture image #1 RGBA source/operand M3D_TEX_SRC_OPD_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE		M3D_TEX_OPD_RGBA_1												M3D_TEX_SRC_RGBA_1[17:16]	
Type			R/W												R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_SRC_RGBA_1[15:0]															
Type	R/W															

M3D_TEX_SRC_RGBA_1 Specifies source of texture image #1
M3D_TEX_OPD_RGBA_1 Specifies operand of texture image #1

M3D_BASE+258h M3D texture image #2 RGBA source/operand M3D_TEX_SRC_OP_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE		M3D_TEX_OPERAND_RGBA_2												M3D_TEX_SRC_RGBA_2[17:16]	
Type			R/W												R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_SRC_RGBA_2[11:0]															
Type	R/W															

M3D_TEX_SRC_RGBA_2 Specifies source of texture image #2
M3D_TEX_OPD_RGBA_2 Specifies operand of texture image #2

M3D_BASE+25Ch M3D fill mode M3D_FILL_MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															M3D_FILL_MODE	
Type															R/W	
Reset															2	

M3D_FILL_MODE Fill mode. (0 represent point, 1 represent wireframe, 2 represent solid)

M3D_BASE+260h|A60h M3D x of visible box lower-left register M3D_VB_XLLC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VB_XLLC[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VB_XLLC [15:0]															
Type	R/W															

M3D_VB_XLLC X of visible box lower-left. (0260h : data format is s15.16, 2260h : data format is s[8].23)

M3D_BASE+264h|A64h M3D y of visible box lower-left register M3D_VB_YLLC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VB_YLLC[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VB_YLLC [15:0]															
Type	R/W															

M3D_VB_YLLC Y of visible box lower-left. (0264h : data format is s15.16, 2264h : data format is s[8].23)

M3D_BASE+268h|A68h M3D y of visible box upper-right register M3D_VB_XURC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VB_XURC[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VB_XURC [15:0]															
Type	R/W															

M3D_VB_XURC X of visible box upper-right. (0268h : data format is s15.16, 2268h : data format is s[8].23)

M3D_BASE+26Ch|A6Ch M3D y of visible box upper-right register M3D_VB_YURC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VB_YURC[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VB_YURC [15:0]															
Type	R/W															

M3D_VB_YURC Y of visible box upper-right. (026Ch : data format is s15.16, 226Ch : data format is s[8].23)

© 2009 MediaTek Inc.

M3D_BASE+270h|A70h M3D frustum near register M3D_FRUSTUM_NEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_FRUSTUM_NEAR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_FRUSTUM_NEAR [15:0]															
Type	R/W															

M3D_FRUSTUM_NEAR n in glFrustum. (0270h : data format is s15.16, 2270h : data format is s[8].23)

M3D_BASE+274h|A74h M3D frustum far register M3D_FRUSTUM_FAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_FRUSTUM_FAR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_FRUSTUM_FAR [15:0]															
Type	R/W															

M3D_FRUSTUM_FAR f in glFrustum. (0274h : data format is s15.16, 2274h : data format is s[8].23)

M3D_BASE+278h|A78h M3D view port near register M3D_VIEWPORT_NEAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VIEWPORT_NEAR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VIEWPORT_NEAR [15:0]															
Type	R/W															

M3D_VIEWPORT_NEAR n in glDepthRange. (0278h : data format is s15.16, 2278h : data format is s[8].23)

M3D_BASE+27Ch|A7Ch M3D view port far register M3D_VIEWPORT_FAR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_VIEWPORT_FAR[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_VIEWPORT_FAR [15:0]															
Type	R/W															

M3D_VIEWPORT_FAR f in glDepthRange. (027Ch : data format is s15.16, 227Ch : data format is s[8].23)

M3D_BASE+280h|A80h M3D model view matrix register M3D_MODEL_VIEW_M_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_1[31:16]															

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_1 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_1 Model view matrix. (0280h : data format is s15.16, 2280h : data format is s[8].23)

M3D_BASE+284h|A84h M3D model view matrix register M3D_MODEL_VIEW_M_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_2 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_2 Model view matrix.

M3D_BASE+288h|A88h M3D model view matrix register M3D_MODEL_VIEW_M_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_3 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_3 Model view matrix.

M3D_BASE+28Ch|A8Ch M3D model view matrix register M3D_MODEL_VIEW_M_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_4 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_4 Model view matrix.

M3D_BASE+290h|A90h M3D model view matrix register M3D_MODEL_VIEW_M_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_5 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_5 Model view matrix.

© 2009 MediaTek Inc.

M3D_BASE+294h|A94h M3D model view matrix register M3D_MODEL_VIEW_M_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_6[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_6 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_6 Model view matrix.

M3D_BASE+298h|A98h M3D model view matrix register M3D_MODEL_VIEW_M_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_7 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_7 Model view matrix.

M3D_BASE+29Ch|A9Ch M3D model view matrix register M3D_MODEL_VIEW_M_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_8 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_8 Model view matrix.

M3D_BASE+2A0h|AA0h M3D model view matrix register M3D_MODEL_VIEW_M_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_9 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_9 Model view matrix.

M3D_BASE+2A4h|AA4h M3D model view matrix register M3D_MODEL_VIEW_M_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_10[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_10 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_10 Model view matrix.

M3D_BASE+2A8h|AA8h M3D model view matrix register M3D_MODEL_VIEW_M_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_11[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_11 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_11 Model view matrix.

M3D_BASE+2ACh|AACh M3D model view matrix register M3D_MODEL_VIEW_M_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_12[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_12 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_12 Model view matrix.

M3D_BASE+2B0h|AB0h M3D model view matrix register M3D_MODEL_VIEW_M_13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_13[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_13 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_13 Model view matrix.

M3D_BASE+2B4h|AB4h M3D model view matrix register M3D_MODEL_VIEW_M_14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_14[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_14 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_14 Model view matrix.

M3D_BASE+2B8h|AB8h **M3D model view matrix register** **M3D_MODEL_VIEW_M_15**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_15[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_15 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_15 Model view matrix.

M3D_BASE+2BCh|ABCh **M3D model view matrix register** **M3D_MODEL_VIEW_M_16**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_MODEL_VIEW_16[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_MODEL_VIEW_16 [15:0]															
Type	R/W															

M3D_MODEL_VIEW_16 Model view matrix.

M3D_BASE+2C0h|AC0h **M3D vp matrix register** **M3D_PV_1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_1 [15:0]															
Type	R/W															

M3D_PV_1 VP matrix. (02C0h : data format is s15.16, 22C0h : data format is s[8].23)

M3D_BASE+2C4h|AC4h **M3D vp matrix register** **M3D_PV_2**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_2 [15:0]															
Type	R/W															

M3D_PV_2 VP matrix.

M3D_BASE+2C8h|AC8h **M3D vp matrix register** **M3D_PV_3**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_3[31:16]															
Type	R/W															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_3 [15:0]															
Type	R/W															

M3D_PV_3 VP matrix.

M3D_BASE+2CCh|ACCh M3D vp matrix register M3D_PV_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_4 [15:0]															
Type	R/W															

M3D_PV_4 VP matrix.

M3D_BASE+2D0h|AD0h M3D vp matrix register M3D_PV_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_5 [15:0]															
Type	R/W															

M3D_PV_5 VP matrix.

M3D_BASE+2D4h|AD4h M3D vp matrix register M3D_PV_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_6[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_6 [15:0]															
Type	R/W															

M3D_PV_6 VP matrix.

M3D_BASE+2D8h|AD8h M3D vp matrix register M3D_PV_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_7 [15:0]															
Type	R/W															

M3D_PV_7 VP matrix.

© 2009 MediaTek Inc.

M3D_BASE+2DCh|ADCh M3D vp matrix register M3D_PV_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_8 [15:0]															
Type	R/W															

M3D_PV_8 VP matrix.

M3D_BASE+2E0h|AE0h M3D vp matrix register M3D_PV_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_9 [15:0]															
Type	R/W															

M3D_PV_9 VP matrix.

M3D_BASE+2E4h|AE4h M3D vp matrix register M3D_PV_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_10[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_10 [15:0]															
Type	R/W															

M3D_PV_10 VP matrix.

M3D_BASE+2E8h|AE8h M3D vp matrix register M3D_PV_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_11[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_11 [15:0]															
Type	R/W															

M3D_PV_11 VP matrix.

M3D_BASE+2ECh|AECh M3D vp matrix register M3D_PV_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_12[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_12 [15:0]															
Type	R/W															

M3D_PV_12 VP matrix.

M3D_BASE+2F0h|AF0h M3D vp matrix register M3D_PV_13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_13[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_13 [15:0]															
Type	R/W															

M3D_PV_13 VP matrix.

M3D_BASE+2F4h|AF4h M3D vp matrix register M3D_PV_14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_14[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_14 [15:0]															
Type	R/W															

M3D_PV_14 VP matrix.

M3D_BASE+2F8h|AF8h M3D vp matrix register M3D_PV_15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_15[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_15 [15:0]															
Type	R/W															

M3D_PV_15 VP matrix.

M3D_BASE+2FCh|AFCh M3D vp matrix register M3D_PV_16

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_PV_16[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_PV_16 [15:0]															
Type	R/W															

M3D_PV_16 VP matrix.

© 2009 MediaTek Inc.

M3D_BASE+300h|B00h M3D user-defined clipping plane matrix register M3D_C_N_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_C_N_X[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_C_N_X [15:0]															
Type	R/W															

M3D_C_N_X User-defined clipping plane matrix. (0300h : data format is s15.16, 2300h : data format is s[8].23)

M3D_BASE+304h|B04h M3D user-defined clipping plane matrix register M3D_C_N_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_C_N_Y[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_C_N_Y [15:0]															
Type	R/W															

M3D_C_N_Y User-defined clipping plane matrix.

M3D_BASE+308h|B08h M3D user-defined clipping plane matrix register M3D_C_N_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_C_N_Z[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_C_N_Z [15:0]															
Type	R/W															

M3D_C_N_Z User-defined clipping plane matrix.

M3D_BASE+30Ch|B0Ch M3D user-defined clipping plane matrix register M3D_C_N_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_C_N_W[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_C_N_W [15:0]															
Type	R/W															

M3D_C_N_W User-defined clipping plane matrix.

M3D_BASE+310h|B10h M3D texture image #0 border RED color M3D_TEX_BORDER_R_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_R_0[31:16]															

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_R_0[15:0]															
Type	R/W															

M3D_TEX_BORDER_R_0 Specifies the border RED color of texture image #0

M3D_BASE+314h|B14h M3D texture image #0 border GREEN color M3D_TEX_BORDER_G_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_G_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_G_0[15:0]															
Type	R/W															

M3D_TEX_BORDER_G_0 Specifies the border GREEN color of texture image #0

M3D_BASE+318h|B18h M3D texture image #0 border BLUE color M3D_TEX_BORDER_B_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_B_0[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_B_0[15:0]															
Type	R/W															

M3D_TEX_BORDER_B_0 Specifies the border BLUE color of texture image #0

M3D_BASE+31Ch|B1Ch M3D texture image #0 border ALPHA color M3D_TEX_BORDER_A_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_A_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_A_1[15:0]															
Type	R/W															

M3D_TEX_BORDER_A_1 Specifies the border ALPHA color of texture image #1

M3D_BASE+320h|B20h M3D texture image #1 border RED color M3D_TEX_BORDER_R_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_R_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_R_1[15:0]															
Type	R/W															

M3D_TEX_BORDER_R_1 Specifies the border RED color of texture image #1

© 2009 MediaTek Inc.

M3D_BASE+324h|B24h M3D texture image #1 border GREEN color M3D_TEX_BORDER_G_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_G_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_G_1[15:0]															
Type	R/W															

M3D_TEX_BORDER_G_1 Specifies the border GREEN color of texture image #1

M3D_BASE+328h|B28h M3D texture image #1 border BLUE color M3D_TEX_BORDER_B_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_B_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_B_1[15:0]															
Type	R/W															

M3D_TEX_BORDER_B_1 Specifies the border BLUE color of texture image #1

M3D_BASE+32Ch|B2Ch M3D texture image #1 border ALPHA color M3D_TEX_BORDER_A_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_A_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_A_1[15:0]															
Type	R/W															

M3D_TEX_BORDER_A_1 Specifies the border ALPHA color of texture image #1

M3D_BASE+330h|B30h M3D texture image #2 border RED color M3D_TEX_BORDER_R_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_R_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_R_2[15:0]															
Type	R/W															

M3D_TEX_BORDER_R_2 Specifies the border RED color of texture image #2

M3D_BASE+334h|B34h M3D texture image #2 border GREEN color M3D_TEX_BORDER_G_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_G_2[31:16]															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_G_2[15:0]															
Type	R/W															

M3D_TEX_BORDER_G_2 Specifies the border GREEN color of texture image #2

M3D_BASE+338h|B38h M3D texture image #2 border BLUE color M3D_TEX_BORDER_B_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_B_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_B_2[15:0]															
Type	R/W															

M3D_TEX_BORDER_B_2 Specifies the border BLUE color of texture image #2

M3D_BASE+33Ch|B3Ch M3D texture image #2 border ALPHA color M3D_TEX_BORDER_A_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_BORDER_A_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_BORDER_A_2[15:0]															
Type	R/W															

M3D_TEX_BORDER_A_2 Specifies the border ALPHA color of texture image #2

M3D_BASE+340h|B40h M3D texture coordinate matrix register M3D_TEX_M_0_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_1 [15:0]															
Type	R/W															

M3D_TEX_M_0_1 Texture coordinate matrix. (0340h : data format is s15.16, 2340h : data format is s[8].23)

M3D_BASE+344h|B44h M3D texture coordinate matrix register M3D_TEX_M_0_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_2 [15:0]															
Type	R/W															

M3D_TEX_M_0_2 Texture coordinate matrix.

M3D_BASE+348h|B48h M3D texture coordinate matrix register M3D_TEX_M_0_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_3 [15:0]															
Type	R/W															

M3D_TEX_M_0_3 Texture coordinate matrix.

M3D_BASE+34Ch|B4Ch M3D texture coordinate matrix register M3D_TEX_M_0_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_4 [15:0]															
Type	R/W															

M3D_TEX_M_0_4 Texture coordinate matrix.

M3D_BASE+350h|B50h M3D texture coordinate matrix register M3D_TEX_M_0_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_5 [15:0]															
Type	R/W															

M3D_TEX_M_0_5 Texture coordinate matrix.

M3D_BASE+354h|B54h M3D texture coordinate matrix register M3D_TEX_M_0_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_6[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_6 [15:0]															
Type	R/W															

M3D_TEX_M_0_6 Texture coordinate matrix.

M3D_BASE+358h|B58h M3D texture coordinate matrix register M3D_TEX_M_0_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_7[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_7 [15:0]															
Type	R/W															

M3D_TEX_M_0_7 Texture coordinate matrix.

M3D_BASE+35Ch|B5Ch M3D texture coordinate matrix register M3D_TEX_M_0_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_8 [15:0]															
Type	R/W															

M3D_TEX_M_0_8 Texture coordinate matrix.

M3D_BASE+360h|B60h M3D texture coordinate matrix register M3D_TEX_M_0_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_9 [15:0]															
Type	R/W															

M3D_TEX_M_0_9 Texture coordinate matrix.

M3D_BASE+364h|B64h M3D texture coordinate matrix register M3D_TEX_M_0_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_10[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_10 [15:0]															
Type	R/W															

M3D_TEX_M_0_10 Texture coordinate matrix.

M3D_BASE+368h|B68h M3D texture coordinate matrix register M3D_TEX_M_0_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_11[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_11 [15:0]															
Type	R/W															

M3D_TEX_M_0_11 Texture coordinate matrix.

M3D_BASE+36Ch|B6Ch M3D texture coordinate matrix register M3D_TEX_M_0_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_0_12[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_0_12 [15:0]															
Type	R/W															

M3D_TEX_M_0_12 Texture coordinate matrix.

M3D_BASE+370h|B70h M3D texture coordinate matrix register M3D_TEX_M_1_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_1 [15:0]															
Type	R/W															

M3D_TEX_M_1_1 Texture coordinate matrix. (0370h : data format is s15.16, 2370h : data format is s[8].23)

M3D_BASE+374h|B74h M3D texture coordinate matrix register M3D_TEX_M_1_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_2 [15:0]															
Type	R/W															

M3D_TEX_M_1_2 Texture coordinate matrix.

M3D_BASE+378h|B78h M3D texture coordinate matrix register M3D_TEX_M_1_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_3 [15:0]															
Type	R/W															

M3D_TEX_M_1_3 Texture coordinate matrix.

M3D_BASE+37Ch|B7Ch M3D texture coordinate matrix register M3D_TEX_M_1_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_4[31:16]															

© 2009 MediaTek Inc.

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_M_1_4 [15:0]**															
Type	R/W															

M3D_TEX_M_1_4 Texture coordinate matrix.

M3D_BASE+380h|B80h M3D texture coordinate matrix register M3D_TEX_M_1_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_M_1_5[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_M_1_5 [15:0]**															
Type	R/W															

M3D_TEX_M_1_5 Texture coordinate matrix.

M3D_BASE+384h|B84h M3D texture coordinate matrix register M3D_TEX_M_1_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_M_1_6[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_M_1_6 [15:0]**															
Type	R/W															

M3D_TEX_M_1_6 Texture coordinate matrix.

M3D_BASE+388h|B88h M3D texture coordinate matrix register M3D_TEX_M_1_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_M_1_7[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_M_1_7 [15:0]**															
Type	R/W															

M3D_TEX_M_1_7 Texture coordinate matrix.

M3D_BASE+38Ch|B8Ch M3D texture coordinate matrix register M3D_TEX_M_1_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_TEX_M_1_8[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_TEX_M_1_8 [15:0]**															
Type	R/W															

M3D_TEX_M_1_8 Texture coordinate matrix.

 © 2009 MediaTek Inc.

M3D_BASE+390h|B90h M3D texture coordinate matrix register M3D_TEX_M_1_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_9 [15:0]															
Type	R/W															

M3D_TEX_M_1_9 Texture coordinate matrix.

M3D_BASE+394h|B94h M3D texture coordinate matrix register M3D_TEX_M_1_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_10[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_10 [15:0]															
Type	R/W															

M3D_TEX_M_1_10 Texture coordinate matrix.

M3D_BASE+398h|B98h M3D texture coordinate matrix register M3D_TEX_M_1_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_11[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_11 [15:0]															
Type	R/W															

M3D_TEX_M_1_11 Texture coordinate matrix.

M3D_BASE+39Ch|B9Ch M3D texture coordinate matrix register M3D_TEX_M_1_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_1_12[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_1_12 [15:0]															
Type	R/W															

M3D_TEX_M_1_12 Texture coordinate matrix.

M3D_BASE+3A0h|BA0h M3D texture coordinate matrix register M3D_TEX_M_2_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_1[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_1 [15:0]															
Type	R/W															

M3D_TEX_M_2_1 Texture coordinate matrix. (03A0h : data format is s15.16, 23A0h : data format is s[8].23)

M3D_BASE+3A4h|BA4h M3D texture coordinate matrix register M3D_TEX_M_2_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_2[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_2 [15:0]															
Type	R/W															

M3D_TEX_M_2_2 Texture coordinate matrix.

M3D_BASE+3A8h|BA8h M3D texture coordinate matrix register M3D_TEX_M_2_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_3 [15:0]															
Type	R/W															

M3D_TEX_M_2_3 Texture coordinate matrix.

M3D_BASE+3ACh|BACh M3D texture coordinate matrix register M3D_TEX_M_2_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_4 [15:0]															
Type	R/W															

M3D_TEX_M_2_4 Texture coordinate matrix.

M3D_BASE+3B0h|BB0h M3D texture coordinate matrix register M3D_TEX_M_2_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_5 [15:0]															
Type	R/W															

M3D_TEX_M_2_5 Texture coordinate matrix.

M3D_BASE+3B4h|BB4h M3D texture coordinate matrix register M3D_TEX_M_2_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_6[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_6 [15:0]															
Type	R/W															

M3D_TEX_M_2_6 Texture coordinate matrix.

M3D_BASE+3B8h|BB8h M3D texture coordinate matrix register M3D_TEX_M_2_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_7 [15:0]															
Type	R/W															

M3D_TEX_M_2_7 Texture coordinate matrix.

M3D_BASE+3BCh|BBCh M3D texture coordinate matrix register M3D_TEX_M_2_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_8 [15:0]															
Type	R/W															

M3D_TEX_M_2_8 Texture coordinate matrix.

M3D_BASE+3C0h|BC0h M3D texture coordinate matrix register M3D_TEX_M_2_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_9 [15:0]															
Type	R/W															

M3D_TEX_M_2_9 Texture coordinate matrix.

M3D_BASE+3C4h|BC4h M3D texture coordinate matrix register M3D_TEX_M_2_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_10[31:16]															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_10 [15:0]															
Type	R/W															

M3D_TEX_M_2_10 Texture coordinate matrix.

M3D_BASE+3C8h|BC8h M3D texture coordinate matrix register M3D_TEX_M_2_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_11[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_11 [15:0]															
Type	R/W															

M3D_TEX_M_2_11 Texture coordinate matrix.

M3D_BASE+3CCh|BCCh M3D texture coordinate matrix register M3D_TEX_M_2_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TEX_M_2_12[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TEX_M_2_12 [15:0]															
Type	R/W															

M3D_TEX_M_2_12 Texture coordinate matrix.

M3D_BASE+3D0h|BD0 D3DM Light Range of Light Source 0 M3D_LIGHT_RANGE_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 0. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3D4h|BD4 D3DM Light Range of Light Source 1 M3D_LIGHT_RANGE_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 1. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3D8h|BD8 D3DM Light Range of Light Source 2 M3D_LIGHT_RANGE_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 2. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3DCh|BDC D3DM Light Range of Light Source 3 M3D_LIGHT_RANGE_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 3. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3E0h|BE0 D3DM Light Range of Light Source 4 M3D_LIGHT_RANGE_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 4. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3E4h|BE4 D3DM Light Range of Light Source 5 M3D_LIGHT_RANGE_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RANGE
Type	R/W

RANGE To set the range of light source 5. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3E8h|BE8 D3DM Light Range of Light Source 6 M3D_LIGHT_RANGE_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 6. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+3ECh|BEC D3DM Light Range of Light Source 7 M3D_LIGHT_RANGE_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RANGE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RANGE															
Type	R/W															

RANGE To set the range of light source 7. When M3D_LIGHT_RANGE_ENABLE is enabled, the attenuation item would be zero if the distance between the vertex and the light source exceeds the range value. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+400h|C00h M3D normal scale register M3D_NORMAL_N_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_1[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_1[15:0]															
Type	R/W															

M3D_NORMAL_N_1 Normal matrix. (0400h : data format is s15.16, 2400h : data format is s[8].23)

M3D_BASE+404h|C04h M3D normal scale register M3D_NORMAL_N_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_2[31:16]															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_2[15:0]															
Type	R/W															

M3D_NORMAL_N_2 Normal matrix.

M3D_BASE+408h|C08h M3D normal scale register M3D_NORMAL_N_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_3[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_3[15:0]															
Type	R/W															

M3D_NORMAL_N_3 Normal matrix.

M3D_BASE+40Ch|C0Ch M3D normal scale register M3D_NORMAL_N_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_4[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_4[15:0]															
Type	R/W															

M3D_NORMAL_N_4 Normal matrix.

M3D_BASE+410h|C10h M3D normal scale register M3D_NORMAL_N_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_5[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_5[15:0]															
Type	R/W															

M3D_NORMAL_N_5 Normal matrix.

M3D_BASE+414h|C14h M3D normal scale register M3D_NORMAL_N_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_6[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_6[15:0]															
Type	R/W															

M3D_NORMAL_N_6 Normal matrix.

M3D_BASE+418h|C18h M3D normal scale register M3D_NORMAL_N_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_7[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_7[15:0]															
Type	R/W															

M3D_NORMAL_N_7 Normal matrix.

M3D_BASE+41Ch|C1Ch M3D normal scale register M3D_NORMAL_N_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_8[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_8[15:0]															
Type	R/W															

M3D_NORMAL_N_8 Normal matrix.

M3D_BASE+420h|C20h M3D normal scale register M3D_NORMAL_N_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_N_9[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_N_9[15:0]															
Type	R/W															

M3D_NORMAL_N_9 Normal matrix.

M3D_BASE+424h|C24h M3D normal scale register M3D_NORMAL_SCALE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_NORMAL_SCALE[31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_NORMAL_SCALE[15:0]															
Type	R/W															

M3D_NORMAL_SCALE Normal will be scaled by this scale factor. (0424h : data format is s15.16, 2424h : data format is s[8].23)

M3D_BASE+428h|C28h Red component of fog color register M3D_FOG_COLOR_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_FOG_COLOR_R[31:16]															

Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_FOG_COLOR_R[15:0]**															
Type	R/W															

M3D_FOG_COLOR_R Red component of fog color. (0428h : data format is s15.16, 2428h : data format is s[8].23)

M3D_BASE+42Ch|C2Ch Green component of fog color register M3D_FOG_COLOR_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_FOG_COLOR_G[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_FOG_COLOR_G[15:0]**															
Type	R/W															

M3D_FOG_COLOR_G Green component of fog color. (042Ch : data format is s15.16, 242Ch : data format is s[8].23)

M3D_BASE+430h|C30h Blue component of fog color register M3D_FOG_COLOR_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_FOG_COLOR_B[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_FOG_COLOR_B[15:0]**															
Type	R/W															

M3D_FOG_COLOR_B Blue component of fog color. (0430h : data format is s15.16, 2430h : data format is s[8].23)

M3D_BASE+43Ch Pixel counter M3D_PXL_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**M3D_PXL_CNT**															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**M3D_PXL_CNT**															
Type	R															

M3D_PXL_CNT Pixel counter

M3D_BASE+440h|C40h Red Component of Ambient Color of Material Register M3D_A_CM_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**R_COLOR**															

Type	R/W

R_COLOR Red componet of a_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+444h|C44h Green Component of Ambient Color of Material Register M3D_A_CM_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+448h|C48h Blue Component of Ambient Color of Material Register M3D_A_CM_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+44Ch Texture cache hit counter M3D_TXCACHE_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_TXCACHE_CNT															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_TXCACHE_CNT															
Type	R															

M3D_TXCACHE_CNT Texture cache hit counter

M3D_BASE+450h|C50h Red Component of Diffusion Color of Material Register M3D_D_CM_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+454h|C54h Green Component of Diffusion Color of Material Register M3D_D_CM_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+458h|C58h Blue Component of Diffusion Color of Material Register M3D_D_CM_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+45Ch|C5Ch Alpha Component of Diffusion Color of Material Register M3D_D_CM_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+460h|C60h Red Component of Specular Color of Material Register M3D_S_CM_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+464h|C64h Green Component of Specular Color of Material Register M3D_S_CM_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+468h|C68h Blue Component of Specular Color of Material Register — M3D_S_CM_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+46Ch|C6Ch Alpha Component of Specular Color of Material Register — M3D_S_CM_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+470h|C70h Red Component of Emission Color of Material Register — M3D_E_CM_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of e_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+474h|C74h Green Component of Emission Color of Material Register — M3D_E_CM_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of e_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+478h|C78h Blue Component of Emission Color of Material Register M3D_E_CM_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of e_{cm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+47Ch Z cache hit counter M3D_ZC_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_ZC_CNT															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_ZC_CNT															
Type	R															

M3D_ZC_CNT Z cache hit counter

M3D_BASE+480h|C80h Red Component of Ambient Intensity of Light Source 0 Register M3D_A_CL_0_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+484h|C84h Green Component of Ambient Intensity of Light Source 0 Register M3D_A_CL_0_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+488h|C88h Blue Component of Ambient Intensity of Light Source 0 Register M3D_A_CL_0_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+48Ch|C8Ch Alpha Component of Ambient Intensity of Light Source 0 Register M3D_A_CL_0_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+490h|C90h Red Component of Ambient Intensity of Light Source 1 Register M3D_A_CL_1_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+494h|C94h Green Component of Ambient Intensity of Light Source 1 Register M3D_A_CL_1_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+498h|C98h Blue Component of Ambient Intensity of Light Source 1 Register M3D_A_CL_1_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+49Ch|C9Ch Alpha Component of Ambient Intensity of Light Source 1 Register M3D_A_CL_1_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ALPHA**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ALPHA**															
Type	R/W															

ALPHA Alpha componet of a_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4A0h|CA0h Red Component of Ambient Intensity of Light Source 2 Register M3D_A_CL_2_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**R_COLOR**															
Type	R/W															

R_COLOR Red componet of a_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4A4h|CA4h Green Component of Ambient Intensity of Light Source 2 Register M3D_A_CL_2_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**G_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**G_COLOR**															
Type	R/W															

G_COLOR Green componet of a_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4A8h|CA8h Blue Component of Ambient Intensity of Light Source 2 Register M3D_A_CL_2_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**B_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**B_COLOR**															
Type	R/W															

B_COLOR Blue componet of a_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4ACh|CACh Alpha Component of Ambient Intensity of Light Source 2 Register M3D_A_CL_2_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ALPHA**															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4B0h|CB0h Red Component of Ambient Intensity of Light Source 3 Register M3D_A_CL_3_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4B4h|CB4h Green Component of Ambient Intensity of Light Source 3 Register M3D_A_CL_3_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4B8h|CB8h Blue Component of Ambient Intensity of Light Source 3 Register M3D_A_CL_3_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4BCh|CBCh Alpha Component of Ambient Intensity of Light Source 3 Register M3D_A_CL_3_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4C0h|CC0h Red Component of Ambient Intensity of Light Source 4 Register M3D_A_CL_4_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4C4h|CC4h Green Component of Ambient Intensity of Light Source 4 Register M3D_A_CL_4_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4C8h|CC8h Blue Component of Ambient Intensity of Light Source 4 Register M3D_A_CL_4_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4CCh|CCCh Alpha Component of Ambient Intensity of Light Source 4 Register M3D_A_CL_4_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4D0h|CD0h Red Component of Ambient Intensity of Light Source 5 Register M3D_A_CL_5_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4D4h|CD4h Green Component of Ambient Intensity of Light Source 5 Register M3D_A_CL_5_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4D8h|CD8h Blue Component of Ambient Intensity of Light Source 5 Register M3D_A_CL_5_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4DCh|CDCh Alpha Component of Ambient Intensity of Light Source 5 Register M3D_A_CL_5_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4E0h|CE0h Red Component of Ambient Intensity of Light Source 6 Register M3D_A_CL_6_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4E4h|CE4h Green Component of Ambient Intensity of Light Source 6 Register — M3D_A_CL_6_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4E8h|CE8h Blue Component of Ambient Intensity of Light Source 6 Register — M3D_A_CL_6_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4ECh|CECh Alpha Component of Ambient Intensity of Light Source 6 Register — M3D_A_CL_6_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4F0h|CF0h Red Component of Ambient Intensity of Light Source 7 Register — M3D_A_CL_7_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4F4h|CF4h Green Component of Ambient Intensity of Light Source 7 Register — M3D_A_CL_7_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4F8h|CF8h Blue Component of Ambient Intensity of Light Source 7 Register M3D_A_CL_7_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+4FCh|CFCh Alpha Component of Ambient Intensity of Light Source 7 Register M3D_A_CL_7_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of a_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+500h|D00h Red Component of Diffusion Intensity of Light Source 0 Register M3D_D_CL_0_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+504h|D04h Green Component of Diffusion Intensity of Light Source 0 Register M3D_D_CL_0_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+508h|D08h Blue Component of Diffusion Intensity of Light Source 0 Register M3D_D_CL_0_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+50Ch|D0Ch Alpha Component of Diffusion Intensity of Light Source 0 Register M3D_D_CL_0_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+510h|D10h Red Component of Diffusion Intensity of Light Source 1 Register M3D_D_CL_1_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+514h|D14h Green Component of Diffusion Intensity of Light Source 1 Register M3D_D_CL_1_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+518h|D18h Blue Component of Diffusion Intensity of Light Source 1 Register M3D_D_CL_1_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+51Ch\|D1Ch Alpha Component of Diffusion Intensity of Light Source 1 Register M3D_D_CL_1_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+520h\|D20h Red Component of Diffusion Intensity of Light Source 2 Register M3D_D_CL_2_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+524h\|D24h Green Component of Diffusion Intensity of Light Source 2 Register M3D_D_CL_2_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+528h\|D28h Blue Component of Diffusion Intensity of Light Source 2 Register M3D_D_CL_2_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+52Ch|D2Ch Alpha Component of Diffusion Intensity of Light Source 2 Register M3D_D_CL_2_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+530h|D30h Red Component of Diffusion Intensity of Light Source 3 Register M3D_D_CL_3_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+534h|D34h Green Component of Diffusion Intensity of Light Source 3 Register M3D_D_CL_3_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+538h|D38h Blue Component of Diffusion Intensity of Light Source 3 Register M3D_D_CL_3_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+53Ch|D3Ch Alpha Component of Diffusion Intensity of Light Source 3 Register M3D_D_CL_3_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+540h|D40h Red Component of Diffusion Intensity of Light Source 4 Register M3D_D_CL_4_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+544h|D44h Green Component of Diffusion Intensity of Light Source 4 Register M3D_D_CL_4_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+548h|D48h Blue Component of Diffusion Intensity of Light Source 4 Register M3D_D_CL_4_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+54Ch|D4Ch Alpha Component of Diffusion Intensity of Light Source 4 Register M3D_D_CL_4_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+550h|D50h Red Component of Diffusion Intensity of Light Source 5 Register M3D_D_CL_5_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**R_COLOR**															
Type	R/W															

R_COLOR Red componet of d_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+554h|D54h Green Component of Diffusion Intensity of Light Source 5 Register M3D_D_CL_5_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**G_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**G_COLOR**															
Type	R/W															

G_COLOR Green componet of d_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+558h|D58h Blue Component of Diffusion Intensity of Light Source 5 Register M3D_D_CL_5_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**B_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**B_COLOR**															
Type	R/W															

B_COLOR Blue componet of d_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+55Ch|D5Ch Alpha Component of Diffusion Intensity of Light Source 5 Register M3D_D_CL_5_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ALPHA**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ALPHA**															
Type	R/W															

ALPHA Alpha componet of d_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+560h|D60h Red Component of Diffusion Intensity of Light Source 6 Register M3D_D_CL_6_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+564h|D64h Green Component of Diffusion Intensity of Light Source 6 Register M3D_D_CL_6_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+568h|D68h Blue Component of Diffusion Intensity of Light Source 6 Register M3D_D_CL_6_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+56Ch|D6Ch Alpha Component of Diffusion Intensity of Light Source 6 Register M3D_D_CL_6_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+570h|D70h Red Component of Diffusion Intensity of Light Source 7 Register M3D_D_CL_7_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of d_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+574h|D74h Green Component of Diffusion Intensity of Light Source 7 Register M3D_D_CL_7_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of d_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+578h|D78h Blue Component of Diffusion Intensity of Light Source 7 Register M3D_D_CL_7_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of d_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+57Ch|D7Ch Alpha Component of Diffusion Intensity of Light Source 7 Register M3D_D_CL_7_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of d_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+580h|D80h Red Component of Specular Intensity of Light Source 0 Register M3D_S_CL_0_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+584h|D84h Green Component of Specular Intensity of Light Source 0 Register M3D_S_CL_0_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+588h\|D88h Blue Component of Specular Intensity of Light Source 0 Register M3D_S_CL_0_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+58Ch\|D8Ch Alpha Component of Specular Intensity of Light Source 0 Register M3D_S_CL_0_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+590h\|D90h Red Component of Specular Intensity of Light Source 1 Register M3D_S_CL_1_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+594h\|D94h Green Component of Specular Intensity of Light Source 1 Register M3D_S_CL_1_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+598h|D98h Blue Component of Specular Intensity of Light Source 1 Register M3D_S_CL_1_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**B_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**B_COLOR**															
Type	R/W															

B_COLOR Blue componet of s_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+59Ch|D9Ch Alpha Component of Specular Intensity of Light Source 1 Register M3D_S_CL_1_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ALPHA**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ALPHA**															
Type	R/W															

ALPHA Alpha componet of s_{cl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5A0h|DA0h Red Component of Specular Intensity of Light Source 2 Register M3D_S_CL_2_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**R_COLOR**															
Type	R/W															

R_COLOR Red componet of s_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5A4h|DA4h Green Component of Specular Intensity of Light Source 2 Register M3D_S_CL_2_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**G_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**G_COLOR**															
Type	R/W															

G_COLOR Green componet of s_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5A8h|DA8h Blue Component of Specular Intensity of Light Source 2 Register M3D_S_CL_2_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**B_COLOR**															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5ACh|DACh Alpha Component of Specular Intensity of Light Source 2 Register M3D_S_CL_2_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5B0h|DB0h Red Component of Specular Intensity of Light Source 3 Register M3D_S_CL_3_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5B4h|DB4h Green Component of Specular Intensity of Light Source 3 Register M3D_S_CL_3_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5B8h|DB8h Blue Component of Specular Intensity of Light Source 3 Register M3D_S_CL_3_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5BCh|DBCh Alpha Component of Specular Intensity of Light Source 3 Register M3D_S_CL_3_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5C0h|DC0h Red Component of Specular Intensity of Light Source 4 Register M3D_S_CL_4_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5C4h|DC4h Green Component of Specular Intensity of Light Source 4 Register M3D_S_CL_4_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5C8h|DC8h Blue Component of Specular Intensity of Light Source 4 Register M3D_S_CL_4_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5CCh|DCCh Alpha Component of Specular Intensity of Light Source 4 Register M3D_S_CL_4_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5D0h|DD0h Red Component of Specular Intensity of Light Source 5 Register M3D_S_CL_5_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5D4h|DD4h Green Component of Specular Intensity of Light Source 5 Register M3D_S_CL_5_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5D8h|DD8h Blue Component of Specular Intensity of Light Source 5 Register M3D_S_CL_5_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5DCh|DDCh Alpha Component of Specular Intensity of Light Source 5 Register M3D_S_CL_5_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

 © 2009 MediaTek Inc.

M3D_BASE+5E0h|DE0h Red Component of Specular Intensity of Light Source 6 Register M3D_S_CL_6_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**R_COLOR**															
Type	R/W															

R_COLOR Red componet of s_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5E4h|DE4h Green Component of Specular Intensity of Light Source 6 Register M3D_S_CL_6_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**G_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**G_COLOR**															
Type	R/W															

G_COLOR Green componet of s_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5E8h|DE8h Blue Component of Specular Intensity of Light Source 6 Register M3D_S_CL_6_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**B_COLOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**B_COLOR**															
Type	R/W															

B_COLOR Blue componet of s_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5ECh|DECh Alpha Component of Specular Intensity of Light Source 6 Register M3D_S_CL_6_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ALPHA**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ALPHA**															
Type	R/W															

ALPHA Alpha componet of s_{cl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5F0h|DF0h Red Component of Specular Intensity of Light Source 7 Register M3D_S_CL_7_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**R_COLOR**															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of s_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5F4h|DF4h Green Component of Specular Intensity of Light Source 7 Register M3D_S_CL_7_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of s_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5F8h|DF8h Blue Component of Specular Intensity of Light Source 7 Register M3D_S_CL_7_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of s_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+5FCh|DFCh Alpha Component of Specular Intensity of Light Source 7 Register M3D_S_CL_7_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALPHA															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ALPHA															
Type	R/W															

ALPHA Alpha componet of s_{cl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+600h|E00h X Coordinate of Position of Light Source 0 Register M3D_P_PL_0_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+604h|E04h Y Coordinate of Position of Light Source 0 Register M3D_P_PL_0_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+608h|E08h Z Coordinate of Position of Light Source 0 Register M3D_P_PL_0_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+60Ch|E0Ch W Coordinate of Position of Light Source 0 Register M3D_P_PL_0_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+610h|E10h X Coordinate of Position of Light Source 1 Register M3D_P_PL_1_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+614h|E14h Y Coordinate of Position of Light Source 1 Register M3D_P_PL_1_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+618h|E18h Z Coordinate of Position of Light Source 1 Register M3D_P_PL_1_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+61Ch|E1Ch W Coordinate of Position of Light Source 1 Register M3D_P_PL_1_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+620h|E20h X Coordinate of Position of Light Source 2 Register M3D_P_PL_2_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+624h|E24h Y Coordinate of Position of Light Source 2 Register M3D_P_PL_2_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+628h|E28h Z Coordinate of Position of Light Source 2 Register M3D_P_PL_2_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+62Ch|E2Ch W Coordinate of Position of Light Source 2 Register M3D_P_PL_2_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+630h|E30h X Coordinate of Position of Light Source 3 Register M3D_P_PL_3_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+634h|E34h Y Coordinate of Position of Light Source 3 Register M3D_P_PL_3_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+638h|E38h Z Coordinate of Position of Light Source 3 Register M3D_P_PL_3_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+63Ch|E3Ch W Coordinate of Position of Light Source 3 Register M3D_P_PL_3_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+640h|E40h X Coordinate of Position of Light Source 4 Register M3D_P_PL_4_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+644h|E44h Y Coordinate of Position of Light Source 4 Register M3D_P_PL_4_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+648h|E48h Z Coordinate of Position of Light Source 4 Register M3D_P_PL_4_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+64Ch|E4Ch W Coordinate of Position of Light Source 4 Register M3D_P_PL_4_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+650h|E50h X Coordinate of Position of Light Source 5 Register M3D_P_PL_5_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+654h|E54h Y Coordinate of Position of Light Source 5 Register M3D_P_PL_5_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+658h|E58h Z Coordinate of Position of Light Source 5 Register M3D_P_PL_5_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+65Ch|E5Ch W Coordinate of Position of Light Source 5 Register M3D_P_PL_5_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+660h|E60h X Coordinate of Position of Light Source 6 Register M3D_P_PL_6_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+664h|E64h Y Coordinate of Position of Light Source 6 Register M3D_P_PL_6_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+668h|E68h Z Coordinate of Position of Light Source 6 Register M3D_P_PL_6_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+66Ch|E6Ch W Coordinate of Position of Light Source 6 Register M3D_P_PL_6_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+670h|E70h X Coordinate of Position of Light Source 7 Register M3D_P_PL_7_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_COORD															
Type	R/W															

X_COORD X coordinate of p_{pl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+674h|E74h Y Coordinate of Position of Light Source 7 Register M3D_P_PL_7_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_COORD															
Type	R/W															

Y_COORD Y coordinate of p_{pl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+678h|E78h Z Coordinate of Position of Light Source 7 Register M3D_P_PL_7_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_COORD															
Type	R/W															

Z_COORD Z coordinate of p_{pl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+67Ch|E7Ch W Coordinate of Position of Light Source 7 Register M3D_P_PL_7_W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	W_COORD															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	W_COORD															
Type	R/W															

W_COORD W coordinate of p_{pl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+680h|E80h X Direction of Spotlight for Light Source 0 Register M3D_S_DL_0_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_DIREC															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_DIREC															
Type	R/W															

X_DIREC X direction of s_{dl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+684h|E84h Y Direction of Spotlight for Light Source 0 Register M3D_S_DL_0_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_DIREC															
Type	R/W															

Y_DIREC Y direction of s_{dl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+688h|E88h Z Direction of Spotlight for Light Source 0 Register M3D_S_DL_0_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_DIREC															
Type	R/W															

Z_DIREC Z direction of s_{dl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+68Ch|E8Ch Spotlight Exponent for Light Source 0 Register M3D_S_RL_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXP															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXP															
Type	R/W															

EXP Spotlight exponent s_{rl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+690h|E90h X Direction of Spotlight for Light Source 1 Register M3D_S_DL_1_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_DIREC															
Type	R/W															

X_DIREC X direction of s_{dl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+694h|E94h Y Direction of Spotlight for Light Source 1 Register M3D_S_DL_1_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y_DIREC**															
Type	R/W															

Y_DIREC Y direction of s_{dl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+698h|E98h Z Direction of Spotlight for Light Source 1 Register M3D_S_DL_1_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Z_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Z_DIREC**															
Type	R/W															

Z_DIREC Z direction of s_{dl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+69Ch|E9Ch Spotlight Exponent for Light Source 1 Register M3D_S_RL_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EXP**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXP**															
Type	R/W															

EXP Spotlight exponent s_{rl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6A0h|EA0h X Direction of Spotlight for Light Source 2 Register M3D_S_DL_2_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X_DIREC**															
Type	R/W															

X_DIREC X direction of s_{dl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6A4h|EA4h Y Direction of Spotlight for Light Source 2 Register M3D_S_DL_2_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y_DIREC**															
Type	R/W															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y_DIREC**															
Type	R/W															

Y_DIREC Y direction of s_{dl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6A8h|EA8h Z Direction of Spotlight for Light Source 2 Register M3D_S_DL_2_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Z_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Z_DIREC**															
Type	R/W															

Z_DIREC Z direction of s_{dl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6ACh|EACh Spotlight Exponent for Light Source 2 Register M3D_S_RL_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EXP**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXP**															
Type	R/W															

EXP Spotlight exponent s_{rl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6B0h|EB0h X Direction of Spotlight for Light Source 3 Register M3D_S_DL_3_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X_DIREC**															
Type	R/W															

X_DIREC X direction of s_{dl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6B4h|EB4h Y Direction of Spotlight for Light Source 3 Register M3D_S_DL_3_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y_DIREC**															
Type	R/W															

Y_DIREC Y direction of s_{dl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6B8h|EB8h Z Direction of Spotlight for Light Source 3 Register M3D_S_DL_3_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_DIREC															
Type	R/W															

Z_DIREC Z direction of s_{dl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6BCh|EBCh Spotlight Exponent for Light Source 3 Register M3D_S_RL_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXP															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXP															
Type	R/W															

EXP Spotlight exponent s_{rl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6C0h|EC0h X Direction of Spotlight for Light Source 4 Register M3D_S_DL_4_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_DIREC															
Type	R/W															

X_DIREC X direction of s_{dl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6C4h|EC4h Y Direction of Spotlight for Light Source 4 Register M3D_S_DL_4_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_DIREC															
Type	R/W															

Y_DIREC Y direction of s_{dl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6C8h|EC8h Z Direction of Spotlight for Light Source 4 Register M3D_S_DL_4_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_DIREC															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Z_DIREC**															
Type	R/W															

Z_DIREC Z direction of s_{dl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6CCh|ECCh Spotlight Exponent for Light Source 4 Register M3D_S_RL_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EXP**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXP**															
Type	R/W															

EXP Spotlight exponent s_{rl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6D0h|ED0h X Direction of Spotlight for Light Source 5 Register M3D_S_DL_5_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X_DIREC**															
Type	R/W															

X_DIREC X direction of s_{dl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6D4h|ED4h Y Direction of Spotlight for Light Source 5 Register M3D_S_DL_5_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y_DIREC**															
Type	R/W															

Y_DIREC Y direction of s_{dl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6D8h|ED8h Z Direction of Spotlight for Light Source 5 Register M3D_S_DL_5_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Z_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Z_DIREC**															
Type	R/W															

Z_DIREC Z direction of s_{dl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+6DCh|EDCh Spotlight Exponent for Light Source 5 Register M3D_S_RL_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXP															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXP															
Type	R/W															

EXP Spotlight exponent s_{rl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6E0h|EE0h X Direction of Spotlight for Light Source 6 Register M3D_S_DL_6_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	X_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	X_DIREC															
Type	R/W															

X_DIREC X direction of s_{dl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6E4h|EE4h Y Direction of Spotlight for Light Source 6 Register M3D_S_DL_6_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Y_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y_DIREC															
Type	R/W															

Y_DIREC Y direction of s_{dl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6E8h|EE8h Z Direction of Spotlight for Light Source 6 Register M3D_S_DL_6_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z_DIREC															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z_DIREC															
Type	R/W															

Z_DIREC Z direction of s_{dl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6ECh|EECh Spotlight Exponent for Light Source 6 Register M3D_S_RL_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXP															
Type	R/W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXP**															
Type	R/W															

EXP Spotlight exponent s_{rl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6F0h|EF0h X Direction of Spotlight for Light Source 7 Register M3D_S_DL_7_X

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**X_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**X_DIREC**															
Type	R/W															

X_DIREC X direction of s_{dl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6F4h|EF4h Y Direction of Spotlight for Light Source 7 Register M3D_S_DL_7_Y

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Y_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Y_DIREC**															
Type	R/W															

Y_DIREC Y direction of s_{dl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6F8h|EF8h Z Direction of Spotlight for Light Source 7 Register M3D_S_DL_7_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Z_DIREC**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Z_DIREC**															
Type	R/W															

Z_DIREC Z direction of s_{dl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+6FCh|EFCh Spotlight Exponent for Light Source 7 Register M3D_S_RL_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**EXP**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXP**															
Type	R/W															

EXP Spotlight exponent s_{rl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+700h|F00h Spotlight Cutoff Angle for Light Source 0 Register M3D_C_RL_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl0}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+704h|F04h Constant Attenuation Factor for Light Source 0 Register M3D_K_0_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{00}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+708h|F08h Linear Attenuation Factor for Light Source 0 Register M3D_K_1_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{10}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+70Ch|F0Ch Quadratic Attenuation Factor for Light Source 0 Register M3D_K_2_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{20}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+710h|F10h Spotlight Cutoff Angle for Light Source 1 Register M3D_C_RL_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COSINE**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COSINE**															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl1}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+714h|F14h Constant Attenuation Factor for Light Source 1 Register M3D_K_0_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{01}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+718h|F18h Linear Attenuation Factor for Light Source 1 Register M3D_K_1_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{11}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+71Ch|F1Ch Quadratic Attenuation Factor for Light Source 1 Register M3D_K_2_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{21}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+720h|F20h Spotlight Cutoff Angle for Light Source 2 Register M3D_C_RL_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COSINE**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COSINE**															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl2}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+724h|F24h Constant Attenuation Factor for Light Source 2 Register M3D_K_0_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{02}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+728h|F28h Linear Attenuation Factor for Light Source 2 Register M3D_K_1_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{12}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+72Ch|F2Ch Quadratic Attenuation Factor for Light Source 2 Register M3D_K_2_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**FACTOR**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FACTOR**															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{22}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+730h|F30h Spotlight Cutoff Angle for Light Source 3 Register M3D_C_RL_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl3}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+734h|F34h Constant Attenuation Factor for Light Source 3 Register M3D_K_0_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{03}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+738h|F38h Linear Attenuation Factor for Light Source 3 Register M3D_K_1_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{13}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+73Ch|F3Ch Quadratic Attenuation Factor for Light Source 3 Register M3D_K_2_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{23}. The format is s15.16 fixed-point or s[8].23 floating-point.

© 2009 MediaTek Inc.

M3D_BASE+740h|F40h Spotlight Cutoff Angle for Light Source 4 Register M3D_C_RL_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl4}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+744h|F44h Constant Attenuation Factor for Light Source 4 Register M3D_K_0_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{04}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+748h|F48h Linear Attenuation Factor for Light Source 4 Register M3D_K_1_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{14}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+74Ch|F4Ch Quadratic Attenuation Factor for Light Source 4 Register M3D_K_2_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{24}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+750h|F50h Spotlight Cutoff Angle for Light Source 5 Register M3D_C_RL_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl5}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+754h|F54h Constant Attenuation Factor for Light Source 5 Register M3D_K_0_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{05}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+758h|F58h Linear Attenuation Factor for Light Source 5 Register M3D_K_1_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{15}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+75Ch|F5Ch Quadratic Attenuation Factor for Light Source 5 Register M3D_K_2_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{25}. The format is s15.16 fixed-point or s[8].23 floating-point.

 © 2009 MediaTek Inc.

M3D_BASE+760h|F60h Spotlight Cutoff Angle for Light Source 6 Register M3D_C_RL_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl6}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+764h|F64h Constant Attenuation Factor for Light Source 6 Register M3D_K_0_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{06}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+768h|F68h Linear Attenuation Factor for Light Source 6 Register M3D_K_1_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{16}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+76Ch|F6Ch Quadratic Attenuation Factor for Light Source 6 Register M3D_K_2_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{26}. The format is s15.16 fixed-point or s[8].23 floating-point.

 © 2009 MediaTek Inc.

M3D_BASE+770h|F70h Spotlight Cutoff Angle for Light Source 7 Register M3D_C_RL_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COSINE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COSINE															
Type	R/W															

COSINE Cosine value of spotlight cutoff angle c_{rl7}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+774h|F74h Constant Attenuation Factor for Light Source 7 Register M3D_K_0_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light constant attenuation factor k_{07}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+778h|F78h Linear Attenuation Factor for Light Source 7 Register M3D_K_1_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light linear attenuation factor k_{17}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+77Ch|F7Ch Quadratic Attenuation Factor for Light Source 7 Register M3D_K_2_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FACTOR															
Type	R/W															

FACTOR Non-directional light quadratic attenuation factor k_{27}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+780h|F80h Red Component of Ambient Color of Scene Register M3D_A_CS_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	R_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_COLOR															
Type	R/W															

R_COLOR Red componet of a_{cs}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+784h|F84h Green Component of Ambient Color of Scene Register M3D_A_CS_G

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	G_COLOR															
Type	R/W															

G_COLOR Green componet of a_{cs}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+788h|F88h Blue Component of Ambient Color of Scene Register M3D_A_CS_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_COLOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_COLOR															
Type	R/W															

B_COLOR Blue componet of a_{cs}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+78Ch Color cache hit counter M3D_CC_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_CC_CNT															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_CC_CNT															
Type	R															

M3D_CC_CNT Color cache hit counter

M3D_BASE+790h|F90h Specular Exponent of Material Register M3D_S_RM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXP															
Type	R/W															

© 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXP															
Type	R/W															

EXP Specular exponent of material s_{rm}. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+794h|F94h Fog Density Register M3D_FOG_DENSITY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DENSITY															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DENSITY															
Type	R/W															

DENSITY Fog density d. The format is s15.16 fixed-point or s[8].23 floating-point. Used when fog mode is EXP or EXP2.

M3D_BASE+798h|F98h Fog Start Register M3D_FOG_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	START															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	START															
Type	R/W															

START Fog start s. The format is s15.16 fixed-point or s[8].23 floating-point. Used when fog mode is LINEAR.

M3D_BASE+79Ch|F9Ch Fog End Register M3D_FOG_END

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	END															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	END															
Type	R/W															

END Fog end e. The format is s15.16 fixed-point or s[8].23 floating-point. Used when fog mode is LINEAR.

M3D_BASE+7A0h|FA0h Lower Bound of Point Size Register M3D_POINT_SIZE_MIN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MIN															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MIN															

Type	R/W
Reset	0

MIN Lower bound of point size. If derived point size is less than MIN, it will be clamped to MIN.

M3D_BASE+7A4h|FA4h Upper Bound of Point Size Register M3D_PONIT_SIZE_MAX

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MAX**															
Type	R/W															
Reset	0x00010000 (s15.16) or 0x3f800000 (s[8].23)															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAX**															
Type	R/W															
Reset	0x00010000 (s15.16) or 0x3f800000 (s[8].23)															

MAX Upper bound of point size. If derived point size is greater than MAX, it will be clamped to MAX.

M3D_BASE+7A8h|FA8h Constant Coefficient of Distance Attenaution Fucntion for Point Size Register M3D_POINT_ATTENUATION_A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COEFF_A**															
Type	R/W															
Reset	0x00010000 (s15.16) or 0x3f800000 (s[8].23)															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COEFF_A**															
Type	R/W															
Reset	0x00010000 (s15.16) or 0x3f800000 (s[8].23)															

COEFF_A Constant coefficient of distance attenaution fucntion to calculate derived point size. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+7ACh|FACh Linear Coefficient of Distance Attenaution Fucntion for Point Size Register M3D_POINT_ATTENUATION_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COEFF_B**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**COEFF_B**															
Type	R/W															
Reset	0															

COEFF_B Linear coefficient of distance attenaution fucntion to calculate derived point size. The format is s15.16 fixed-point or s[8].23 floating-point.

M3D_BASE+7B0h|FB0h Quadratic Coefficient of Distance Attenaution Fucntion for Point Size Register M3D_POINT_ATTENUATION_C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**COEFF_C**															
Type	R/W															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COEFF_C															
Type	R/W															
Reset	0															

COEFF_C Quadratic coefficient of distance attenaution fucntion to calculate derived point size. The format is s15.16 fixed-point or s[8].23 floating-point

M3D_BASE+7B4h M3D texture image #0 filter/wrap mode M3D_TEX_PARA_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVE						M3D_TEX_WRAP_T_0			M3D_TEX_WRAP_S_0			M3D_TEX_MAG_FILTER_0	M3D_TEX_MIN_FILTER_0		
Type							R/W			R/W			R/W	R/W		

M3D_TEX_MIN_FILTER_0 Specifies minify filter mode of texture image #0

000	M3D_TEX_NEAREST
001	M3D_TEX_LINEAR
010	M3D_TEX_NEAREST_MIPMAP_NEAREST
011	M3D_TEX_LINEAR_MIPMAP_NEAREST
100	M3D_TEX_NEAREST_MIPMAP_LINEAR
101	M3D_TEX_ LINEAR _MIPMAP_ LINEAR

M3D_TEX_MAG_FILTER_0 Specifies magnify filter mode of texture image #0

0	M3D_TEX_NEAREST
1	M3D_TEX_LINEAR

M3D_TEX_WRAP_S_0 Specifies S-direction wrap mode of texture image #0

0	M3D_TEX_WRAP_S_REPEAT
1	M3D_TEX_WRAP_S_CLAMP_TO_EDGE
2	M3D_TEX_WRAP_S_MIRROR
3	M3D_TEX_WRAP_S_CLAMP
4	M3D_TEX_WRAP_S_BORDER

M3D_TEX_WRAP_T_0 Specifies T-direction wrap mode of texture image #0

M3D_BASE+7B8h M3D texture image #1 filter/wrap mode M3D_TEX_PARA_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RESERVE						M3D_TEX_WRAP_T_1			M3D_TEX_WRAP_S_1			M3D_TEX_MAG_FILTER_1	M3D_TEX_MIN_FILTER_1		
Type							R/W			R/W			R/W	R/W		

M3D_TEX_MIN_FILTER_1	Specifies minify filter mode of texture image #1
M3D_TEX_MAG_FILTER_1	Specifies magnify filter mode of texture image #1
M3D_TEX_WRAP_S_1	Specifies S-direction wrap mode of texture image #1
M3D_TEX_WRAP_T_1	Specifies T-direction wrap mode of texture image #1

M3D_BASE+7BCh M3D texture image #2 filter/wrap mode M3D_TEX_PARA_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVE						M3D_TEX_WRAP_T_2			M3D_TEX_WRAP_S_2			M3D_TEX_MAG_FILTER_2	M3D_TEX_MIN_FILTER_2		
Type							R/W			R/W			R/W	R/W		

M3D_TEX_MIN_FILTER_2	Specifies minify filter mode of texture image #2
M3D_TEX_MAG_FILTER_2	Specifies magnify filter mode of texture image #2
M3D_TEX_WRAP_S_2	Specifies S-direction wrap mode of texture image #2
M3D_TEX_WRAP_T_2	Specifies T-direction wrap mode of texture image #2

M3D_BASE+7C0h M3D cache performance counter control M3D_CACHE_PERF_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE															
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVE						M3D_VC_CNT_RST	M3D_VC_CNT_EN	M3D_CC_CNT_RST	M3D_CC_CNT_EN	M3D_ZC_CNT_RST	M3D_ZC_CNT_EN	M3D_TXCACHE_CNT_RST	M3D_TXCACHE_CNT_EN	M3D_PXLCNT_RST	M3D_PXLCNT_EN
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

M3D_PXLCNT_EN	Enable pixel counter
M3D_PXLCNT_RST	Reset pixel counter
M3D_TXCACHE_CNT_EN	Enable texture cache hit counter
M3D_TXCACHE_CNT_RST	Reset texture cache hit counter
M3D_ZC_CNT_EN	Enable z cache hit counter

M3D_ZC_CNT_RST	Reset z cache hit counter
M3D_CC_CNT_EN	Enable color cache hit counter
M3D_CC_CNT_RST	Reset color cache hit counter
M3D_VC_CNT_EN	Enable vertex cache hit counter
M3D_VC_CNT_RST	Reset vertex cache hit counter

M3D_BASE+7C8h M3D drawtex cropper region u coord M3D_DRAWTEX_CRU_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CRU_0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CRU_0															
Type	R/W															

M3D_DRAWTEX_CRU_0 Specifies drawtex cropper region u coord

M3D_BASE+7CCh M3D drawtex cropper region v coord M3D_DRAWTEX_CRV_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CRV_0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CRV_0															
Type	R/W															

M3D_DRAWTEX_CRV_0 Specifies drawtex cropper region v coord

M3D_BASE+7D0h M3D drawtex cropper region delta u coord M3D_DRAWTEX_DCRU_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DCRU_0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DCRU_0															
Type	R/W															

M3D_DRAWTEX_DCRU_0 Specifies drawtex cropper region delta u coord

M3D_BASE+7D4h M3D drawtex cropper region delta v coord M3D_DRAWTEX_DCRV_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DCRV_0															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DCRV_0															
Type	R/W															

M3D_DRAWTEX_DCRV_0 Specifies drawtex cropper region delta v coord

 © 2009 MediaTek Inc.

M3D_BASE+7D8h M3D drawtex cropper region xy coord M3D_DRAWTEX_XY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE												Y			
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Y						X									
Type	R/W															

M3D_DRAWTEX_XY Specifies drawtex cropper region xy coord

M3D_BASE+7DCh M3D drawtex cropper region width/height M3D_DRAWTEX_WH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVE												HEIGHT			
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HEIGHT						WIDTH									
Type	R/W															

M3D_DRAWTEX_WH Specifies drawtex cropper region width/height

M3D_BASE+7E0h|FE0 M3D drawtex cropper region z value M3D_DRAWTEX_Z

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Z															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Z															
Type	R/W															

M3D_DRAWTEX_Z Specifies drawtex cropper region z value. The format is fix s11.16 or floating s[8]23

M3D_BASE+7E4h|FE4 M3D drawtex cropper region fog factor M3D_DRAWTEX_FOG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FOG FACTOR															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FOG FACTOR															
Type	R/W															

M3D_DRAWTEX_FOG Specifies drawtex cropper region fog factor. The format is fix s11.16 or floating s[8]23

M3D_BASE+7E8h M3D dummy register 0 M3D_DUMMY_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DUMMY_0[31:16]															
Type	W															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DUMMY_0[15:0]															
Type	W															

M3D_DUMMY_0 M3D engine dummy register #0

M3D_BASE+7ECh M3D dummy register 1 M3D_DUMMY_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DUMMY_1[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DUMMY_1[15:0]															
Type	R															

M3D_DUMMY_1 M3D engine dummy register #1

M3D_BASE+7F0h M3D dummy register 2 M3D_DUMMY_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DUMMY_2[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DUMMY_2[15:0]															
Type	R															

M3D_DUMMY_2 M3D engine dummy register #2

M3D_BASE+7F4h M3D debug port 5 M3D_DBGRD_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_5[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_5[15:0]															
Type	R															

M3D_DBGRD_5 M3D engine debug port #5 (read only).

M3D_BASE+7F8h M3D debug port 6 M3D_DBGRD_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_6[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_6[15:0]															
Type	R															

M3D_DBGRD_6 M3D engine debug port #6 (read only).

M3D_BASE+7FCh **M3D debug port 7** **M3D_DBGRD_7**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	M3D_DBGRD_7[31:16]															
Type	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	M3D_DBGRD_7[15:0]															
Type	R															

M3D_DBGRD_7 M3D engine debug port #7 (read only).

5.21.3 Application Note

1	M3D_FRAME_BUF_FORMAT	There is no alpha buffer in RGB565 and RGB888 format. Default value 0xFF is used as destination alpha color for blending operation. If application prefers destination alpha for blending op, ARGB8888 mode is recommended.
2	M3D_EARLY_Z_ENABLE	EarlyZ must be disable when alpha test/stencil test/line/point is enable Once EarlyZ is disable at some list, it must be disable until the frame buffer is flush
3	M3D_EARLY_Z_BUF_ADDR	For WVGA size, EarlyZ buffer is 102.48KB(EarlyZ mode 0) or 51.24KB(EarlyZ mode 1), For VGA size, EarlyZ buffer is 76.8KB(EarlyZ mode 0) or 38.4KB(EarlyZ mode 1). For QVGA size, EarlyZ buffer is 19.2KB(EarlyZ mode 0) or 9.6KB(EarlyZ mode 1). If EarlyZ is enable, clear depth buffer and EarlyZ buffer at the same time. The clear value of EarlyZ buffer is {1'b0,M3D_DEPTH_CLEAR_VAL[15:9]}
4	M3D_EARLYZ_MODE	Set m3d_earlyz_mode=0, better performance but more SYSRAM usage Set m3d_earlyz_mode=1, worse performance but less SYSRAM usage If SYSRAM is big enough, set m3d_earlyz_mode 0 is recommended
5	M3D_FRAME_BUF_ADDR	Must be 8-byte alignment
6	M3D_DEPTH_BUF_ADDR	Must be 8-byte alignment
7	M3D_TEX_IMG_PTR_n_n	Must be 8-byte alignment

5.22 MPEG-4 Deblocking Filters

5.22.1 General Description

The purpose of deblocking filter is to reduce blocking artifacts while keeping image edge intact. The filter operations are performed across 8 × 8 block edge boundaries of decoded frames. In our design, deblocking edge filter algorithm in ANNEX J.3 of ITU-T Rec. H.263 is used. The filtering operations include an additional clipping to ensure that resulting pixel values stay in the range 0...255. No filtering is performed across a picture edge. Chrominance as well as luminance data are filtered.

The deblocking filter operates using a set of four pixel values on a horizontal or vertical line of the reconstructed picture, denoted as A, B, C and D, of which A and B belong to one block called block1 and C and D belong to a neighbouring block called block2 which is to the right of or below block1. Filtering across a horizontal edge shall not have been influenced by previous filtering across a vertical edge. In other words, before filtering across a vertical edge using pixels (A, B, C, D), all modifications of pixels (A, B, C, D) resulting from filtering across a horizontal edge shall have taken place. Figure 1.1 shows examples for the position of these pixels.

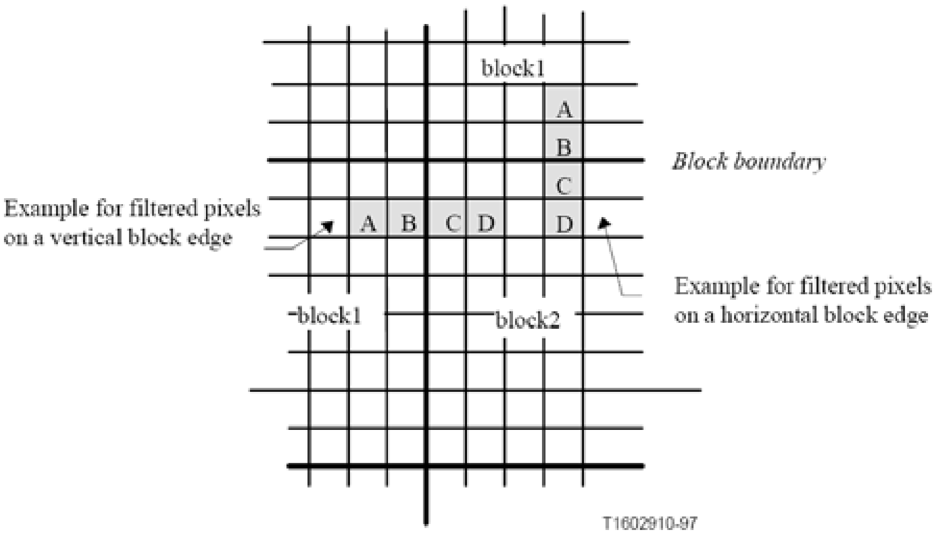


Figure 1.1– Examples of positions of filtered pixels

MP4 debloking filters are post-processing of MP4 decoder and the design is in the image datapath. Its inputs are from ITR0 and multi-ouputs are to CRZ, PRZ and IPP1. ITR0 inputs YUV4:4:4 pixel data to mp4_deblk. Mp4_deblk supports output deblocking pixel data (YUV4:4:4) to three engines simultaneously. The interface of mp4_deblk is shown in Figure 1.2.

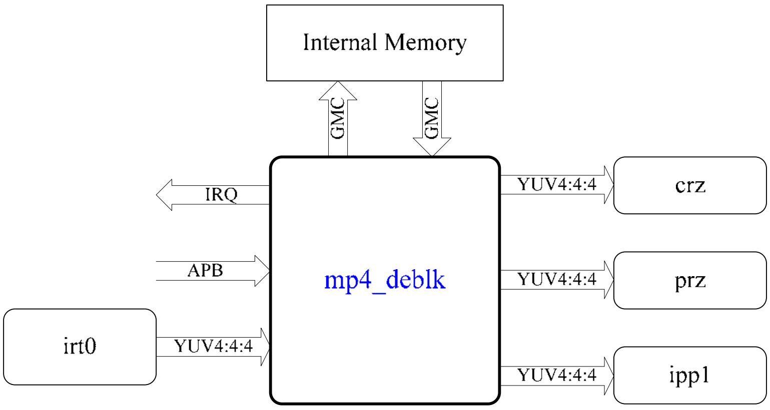


Figure 1.2– Interface of mp4_deblk

The block diagram of mp4_deblk is shown in Figure 1.3.

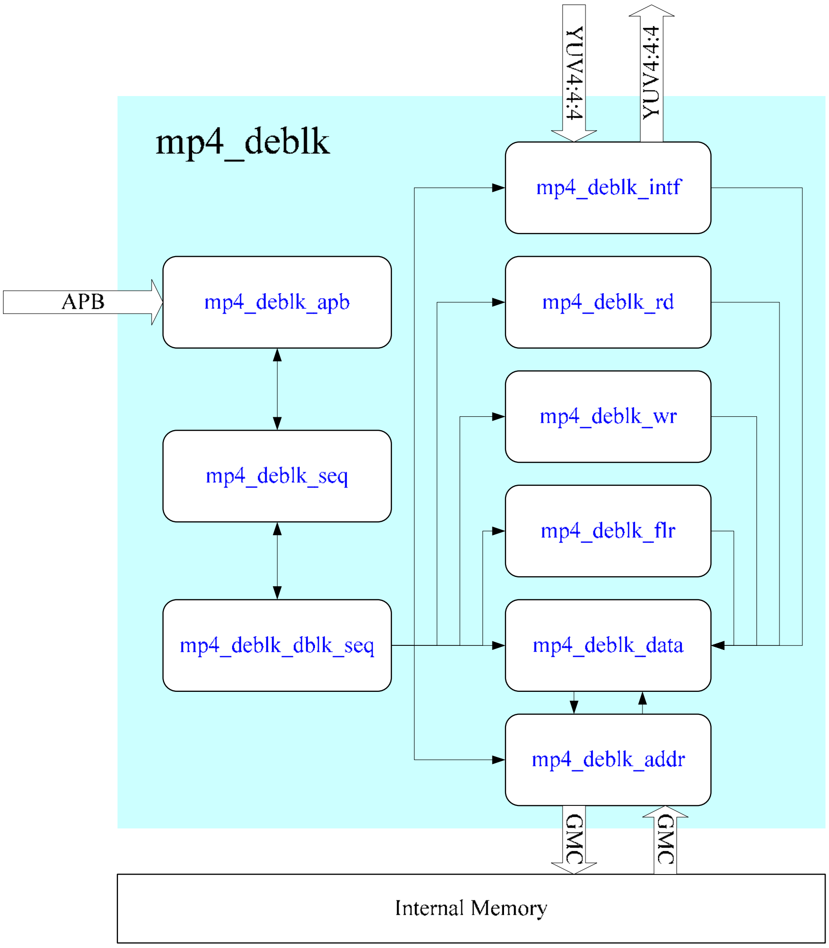


Figure 1.3– Block diagram of mp4_deblk

© 2009 MediaTek Inc.

1.1.2 Register Definitions

REGISTER ADDRESS	REGISTER NAME	ACRONYM
MP4_DEBLK+ 0000h	MP4 DEBLK Command Register	**MP4_DEBLK_COMD**
MP4_DEBLK + 0004h	MP4 DEBLK Configuration Register	**MP4_DEBLK_CONF**
MP4_DEBLK + 0008h	MP4 DEBLK Status Register	**MP4_DEBLK_STS**
MP4_DEBLK + 000Ch	MP4 DEBLK Interrupt Status Register	**MP4_DEBLK_IRQ_STS**
MP4_DEBLK + 0010h	MP4 DEBLK Interrupt Acknowledge Register	**MP4_DEBLK_IRQ_ACK**
MP4_DEBLK + 0014h	MP4 DEBLK Line Buffer Address Register	**MP4_DEBLK_LIN_BUF_ADDR**
MP4_DEBLK + 0018h	MP4 DEBLK Quantizer Address Register	**MP4_DEBLK_QUANT_ADDR**

1.1.2.1 Main Control

MP4_DEBLK+ 0000h **MP4 DEBLK Command Register** **MP4_DBLK_COMD**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	DBLK_START	APB_RST	CORE_RST
Type														WO	WO	WO

This register is the main command register for mp4 deblocking filters.

CORE_RST Software reset control. Writing 1 to this bit will reset the hardware core excluding APB control register set.

APB_RST Software reset control. Writing 1 to this bit will reset the APB control register set of deblocking. Note that this bit won't reset the hardware core.

DBLK_START Start the deblocking operation if writing 1 to this bit and if DBLK_EN is set.

1.1.2.2 Configuration

MP4_DEBLK+ 0004h **MP4 DEBLK Configuration Register** **MP4_DBLK_CONF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	IPP1_EN	PRZ_EN	CRZ_EN	MB_Y_LIMIT							
Type						R/W	R/W	R/W	R/W							
Reset	0	0	0	0	0	0	0	0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MB_X_LIMIT								-	-	-	FLIP	ROTATE		DBLK_EN	IRQ
Type	R/W											R/W	R/W	R/W	R/W	R/W
Reset	0								0	0	0	0	0	0	0	0

This register is used to configure the operating conditions and modes of mp4 deblocking filters.

IRQ Control for interrupt request
- **0** Disable the interrupt reporting mechanism
- 1 Enable the interrupt reporting mechanism

DBLK_EN Deblocking filters
- **0** Disable deblocking filters Operation
- 1 Enable deblocking filters Operation

ROTATE Control for video rotate angle. Need IRT0 DMA to rotate the video for MP4_DEBLK.
- **00** 0°
- **01** Clockwise 90°
- **10** Clockwise 180°
- **11** Clockwise 270°

FLIP Control for flip video. Need IRT0 DMA to flip the video for MP4_DEBLK.
- **0** No flip
- **1** Flip

MB_X_LIMIT Video Frame Width (before rotate/flip). If MB_X_LIMIT is not integer (the decoded frame width is not multiple of 16), round up to integer.

MB_Y_LIMIT Video Frame Height (before rotate/flip). If MB_Y_LIMIT is not integer (the decoded frame height is not multiple of 16), round up to integer.

CRZ_EN Control of output to CRZ
- **0** Disable
- **1** Enable

PRZ_EN Control of output to PRZ
- **0** Disable
- **1** Enable

IPP1_EN Control of output to IPP1
- **0** Disable
- **1** Enable

MP4_DEBLK +0008h MP4 DEBLK Status Register MP4_DBLK_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												STATE				
Type												RO	RO	RO	RO	RO

This register provides the state information of deblocking sequencer for software program. It is a mirror of the HW deblocking sequencer state machine and can be used for debugging.

MP4_DEBLK+000Ch MP4 DEBLK Interrupt Status Register MP4_DBLK_IRQ_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DEBLK_DONE
Type																RO

This register allows software program to poll which interrupt source generates the interrupt request. A bit set to '1' indicates a corresponding active interrupt source. Note that IRQ control bit in MP4_DBLK_CONF should be enabled first in order to activate the interrupt reporting mechanism.

DEBLK_DONE Deblocking filter complete.

MP4_DEBLK+0010h MP4 DEBLK Interrupt Acknowledge Register MP4_DBLK_IRQ_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DEBLK_DONE
Type																WC

This register provides a mean for software program to acknowledge the interrupt source. Writing a '1' to the specific bit position will result in an acknowledgement to the corresponding interrupt source and clear the corresponding bit in MP4_DBLK_IRQ_STS.

DEBLK_DONE Deblocking Filters Task Complete

MP4_DEBLK+0014h MP4 DEBLK Line Buffer Address Register MP4_DBLK_LIN_BUF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LIN_BUF_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LIN_BUF_ADDR														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

This register describes the starting address of internal line buffer. Note that this base address should be 4-byte aligned. And the required internal line buffer size (YUV420 format) should be: 1.5 * number of pixels of frame_width * 8 = 12 * number of pixels of frame_width (bytes).

MP4_DEBLK+0018h MP4 DEBLK Quantizer Address Register MP4_DBLK_QUANT_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	QUANT_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	QUANT_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register describes the starting address of quantizer scales for flip/rotate frame. Note that this base address should be byte-aligned. The base address is dependent on MB_X_LIMIT, MB_Y_LIMIT, FLIP and ROTATE settings. BASE_QUANT_ADDR stores quantizer scales of top-left 4 macroblock in a decoded frame. The table below is the conclusion. (**MB_X** = MB_X_LIMIT round up to multiple of 4.)

FLIP	ROTATE	QUANT_ADDR
0	00	BASE_QUANT_ADDR
0	01	BASE_QUANT_ADDR ＋ **MB_X***(MB_Y_LIMIT －1)
0	10	BASE_QUANT_ADDR ＋ **MB_X***(MB_Y_LIMIT －1)＋(MB_X_LIMIT －1)
0	11	BASE_QUANT_ADDR ＋MB_X_LIMIT－1
1	00	BASE_QUANT_ADDR ＋MB_X_LIMIT－1
1	01	BASE_QUANT_ADDR
1	10	BASE_QUANT_ADDR ＋ **MB_X***(MB_Y_LIMIT －1)
1	11	BASE_QUANT_ADDR ＋ **MB_X***(MB_Y_LIMIT －1)＋(MB_X_LIMIT －1)

5.23 MPEG-4/H.263 Video CODEC

5.23.1 General Description

MPEG-4 is an emerging video coding standard defined in ISO/IEC 14496-2. It is designed to cover a wide range of bit-rates (typically, 5 kbps to 10Mbps). MPEG-4 standard has become one of the enabling factors for mobile multimedia communications. H.263 is another video coding standard that is developed by ITU-T/SG 15 for low-bit-rate applications below 64kbps. H.263 profile 0 level 10 is the mandatory video decoder in 3GPP specification. Therefore, our goal is to design a video codec suited to both MPEG-4 and H.263 standard.

There are two coding modes in MPEG-4 video compression: Intra-frame coding and Inter-frame coding. Intra-frame coding refers to video coding techniques that achieve compression by exploiting the high spatial correlation between neighboring pels within a video frame. Such techniques are also known as spatial

redundancy reduction techniques or still-image coding techniques. Inter-frame coding refers to video coding techniques that achieve compression by exploiting the high temporal correlation between the frames of a video sequence. Such methods are also known as temporal redundancy reduction techniques. Note that inter-frame coding may not be appropriate for some applications. For example, it would be necessary to decode the complete inter-frame coded sequence before being able to randomly access individual frames. Thus, a combined approach is normally used in which a number of frames are intra-frame coded (I-frames) at specific intervals within the sequence and the other frames are inter-frame coded (Predicted or P-frames) with reference to those key frames. Moreover, intra-frame coding is allowed in P-frames.

The ISO/IEC 14496 specification is intended to be generic in the sense that it serves a wide range of applications, bit-rates, resolutions, qualities and services. A number of coding tools are defined in the specification. Considering the practicality of implementing the full syntax of this specification, a limited number of subsets of the syntax are also stipulated by means of “profile” and “level”. A “profile” is a defined subset of the entire bitstream syntax that is defined by this specification. A “level” is a defined set of constraints imposed on parameters in the bitstream. Our application is focused on handset devices. Due to restriction of limited resource, only simple profile is supported for most of handset devices. According to 3GPP TS 26.234 specification, H.263 profile 0 level 10 is the mandatory video decoder. MPEG-4 visual simple profile level 0 is an optional video decoder. The MPEG-4/H.263 codec supports both MPEG-4 simple profile and H.263 baseline profile. Generally, the file extension of MPEG-4 video file is .mp4. The file extension of 3GPP video file is .3gp.

The design implements both decoder and encoder. The decoder block diagram is shown in Figure 142. The encoder block diagram is shown in Figure 143

The decoder specification is as follows:

1. Support ISO/IEC 14496-2 MPEG-4 simple profile @ level 0~3
2. Support H.263 profile 0 level 10 (baseline profile)
3. The following visual tools are supported
    - I-VOP
    - P-VOP
    - AC/DC Prediction
    - 4-MV
    - Unrestricted MV
    - Error Resilience
        - Slice Resynchronization
        - Data Partitioning
        - Reversible VLC
    - Short Header Mode
    - Full and Half Pel accuracy
    - *fcode* can be 1~7

- Maximum horizontal luminance pixel resolution can be up to 720
- Maximum vertical luminance pixel resolution can be up to 480
- Error Concealment
- Single object

The encoder specification is as follows:

1. Support ISO/IEC 14496-2 MPEG-4 simple profile @ level 0, partially support MPEG-4 simple profile @ level 1
2. Support H.263 profile 0 level 10
3. The following visual tools are supported
   - I-VOP
   - P-VOP
   - DC Prediction
   - Unrestricted MV
   - Short Header Mode
   - Full and Half Pel motion estimation
   - Decision making logic
   - *fcode* can be 1~3
   - *intra_dc_vlc_threshold* shall be 0
   - Maximum horizontal luminance pixel resolution can be up to 720
   - Maximum vertical luminance pixel resolution can be up to 480

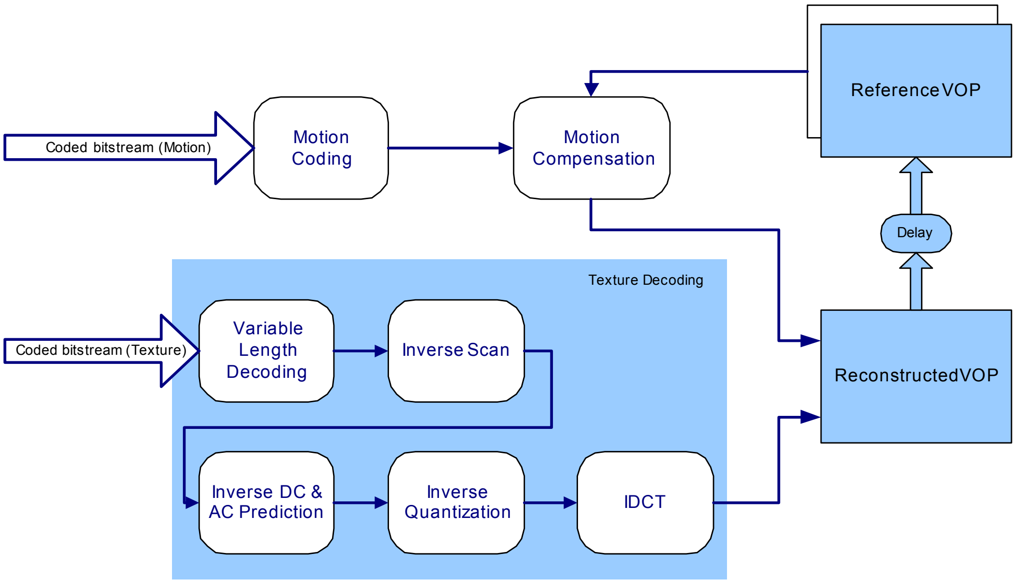


Figure 142 Block Diagram of Decoder

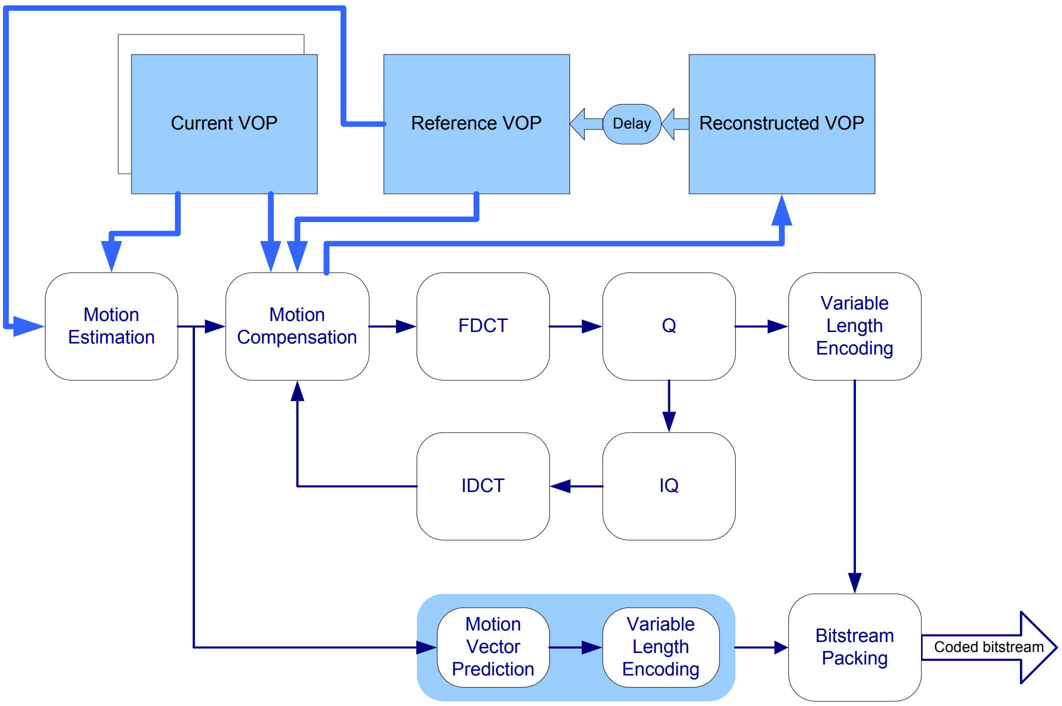


Figure 143 Block Diagram of Encoder

5.23.2 Register Definitions

5.23.2.1 Register Map

REGISTER ADDRESS	REGISTER NAME	SYNONYM
MP4 + 0000h	Video CODEC Command Register	**MP4_CODEC_COMD**
MP4 + 0004h	VLC DMA Command Register	**MP4_VLC_DMA_COMD**
MP4 + 0100h	Encoder Configuration Register	**MP4_ENC_CODEC_CONF**
MP4 + 0104h	Encoder Status Register	**MP4_ENC_STS**
MP4 + 0108h	Encoder Interrupt Mask Register	**MP4_ENC_IRQ_MASK**
MP4 + 010Ch	Encoder Interrupt Status Register	**MP4_ENC_IRQ_STS**
MP4 + 0110h	Encoder Interrupt Acknowledge Register	**MP4_ENC_IRQ_ACK**
MP4 + 0114h	Encoder Configuration Register	**MP4_ENC_CONF**
MP4 + 0124h	Encoder Current VOP Base Address Register	**MP4_ENC_VOP_ADDR**
MP4 + 0128h	Encoder Reference VOP Base Address Register	**MP4_ENC_REF_ADDR**
MP4 + 012Ch	Encoder Reconstructed VOP Base Address Register	**MP4_ENC_REC_ADDR**
MP4 + 0130h	VLE Data Load-Store LSB Base Address Register	**MP4_ENC_DATA_STORE_ADDR**
MP4 + 0134h	DC/AC Prediction Storage LSB Base Address Register	**MP4_ENC_DACP_ADDR**
MP4 + 0138h	Motion Vector Storage LSB Base Address Register	**MP4_ENC_MVP_ADDR**
MP4 + 0140h	Encoder VOP Structure 0 Register	**MP4_ENC_VOP_STRUCT0**
MP4 + 0144h	Encoder VOP Structure 1 Register	**MP4_ENC_VOP_STRUCT1**
MP4 + 0148h	Encoder VOP Structure 2 Register	**MP4_ENC_VOP_STRUCT2**
MP4 + 014Ch	VOP Structure 3 Register	**MP4_VOP_STRUCT3**
MP4 + 0150h	MB Structure 0 Register	**MP4_ENC_MB_STRUCT0**
MP4 + 0160h	Encoder VLC DMA Base Address Register	**MP4_ENC_VLC_BASE_ADDR**
MP4 + 0164h	Encoder VLC DMA Base Bit Count Register	**MP4_ENC_VLC_BASE_BITCNT**
MP4 + 0168h	Encoder VLC DMA Buffer Limit Register	**MP4_ENC_VLC_LIMIT**
MP4 + 016Ch	Encoder VLC DMA Current Word Register	**MP4_ENC_VLC_WORD**
MP4 + 0170h	Encoder VLC DMA Current Bit Count Register	**MP4_ENC_VLC_BITCN**
MP4 + 0174h	Encoder VLC DMA Ring Buffer Ending Address Register	**MP4_ENC_VLC_JUMP_FROM_ADDR**
MP4 + 0178h	Encoder VLC DMA Ring Buffer Starting Address Register	**MP4_ENC_VLC_JUMP_TO_ADDR**
MP4 + 0180h	MPEG4 Encoder Resync Marker Configuration 0 Register	**MP4_ENC_RESYNC_CONF0**
MP4 + 0184h	MPEG4 Encoder Resync Marker Configuration 1 Register	**MP4_ENC_RESYNC_CONF1**
MP4 + 0188h	MPEG4 Encoder Local Time Base Register	**MP4_ENC_TIME_BASE**
MP4 + 018Ch	MPEG4 Encoder Pre-fetch Mode Internal REF Base Address	**MP4_ENC_REF_INT_ADDR**
MP4 + 0190h	MPEG4 Encoder Pre-fetch Mode Internal CUR Base Address	**MP4_ENC_CUR_INT_ADDR**

© 2009 MediaTek Inc.

MP4 + 0194h	MPEG4 Encoder Cycle Count	**MP4_ENC_CYCLE_COUNT**
MP4 + 0200h	Decoder Configuration Register	**MP4_DEC_CODEC_CONF**
MP4 + 0204h	Decoder Status Register	**MP4_DEC_STS**
MP4 + 0208h	Decoder Interrupt Mask Register	**MP4_DEC_IRQ_MASK**
MP4 + 020Ch	Decoder Interrupt Status Register	**MP4_DEC_IRQ_STS**
MP4 + 0210h	Decoder Interrupt Acknowledge Register	**MP4_DEC_IRQ_ACK**
MP4 + 0224h	Decoder Reference VOP Base Address Register	**MP4_DEC_REF_ADDR**
MP4 + 0228h	Decoder Reconstructed VOP Base Address Register	**MP4_DEC_REC_ADDR**
MP4 + 0230h	Decoder Data Load-Store LSB Base Address Register	**MP4_DEC_DATA_STORE_ADDR**
MP4 + 0234h	DC/AC Prediction Storage LSB Base Address Register	**MP4_DEC_DACP_ADDR**
MP4 + 0238h	Motion Vector Storage LSB Base Address Register	**MP4_DEC_MVP_ADDR**
MP4 + 0240h	Decoder VOP Structure 0 Register	**MP4_DEC_VOP_STRUCT0**
MP4 + 0244h	Decoder VOP Structure 1 Register	**MP4_DEC_VOP_STRUCT1**
MP4 + 0248h	Decoder VOP Structure 2 Register	**MP4_DEC_VOP_STRUCT2**
MP4 + 014Ch	Decoder MB Structure 0 Register	**MP4_DEC_MB_STRUCT0**
MP4 + 0260h	Decoder VLC DMA Base Address Register	**MP4_DEC_VLC_BASE_ADDR**
MP4 + 0264h	Decoder VLC DMA Base Bit Count Register	**MP4_DEC_VLC_BASE_BITCNT**
MP4 + 0268h	Decoder VLC DMA Buffer Limit Register	**MP4_DEC_VLC_LIMIT**
MP4 + 026Ch	Decoder VLC DMA Current Word Register	**MP4_DEC_VLC_WORD**
MP4 + 0270h	Decoder VLC DMA Current Bit Count Register	**MP4_DEC_VLC_BITCNT**
MP4 + 0274h	Decoder VLC DMA Ring Buffer Ending Address Register	**MP4_DEC_VLC_JUMP_FROM_A DDR**
MP4 + 0278h	Decoder VLC DMA Ring Buffer Starting Address Register	**MP4_DEC_VLC_JUMP_TO_ADD R**
MP4 + 027Ch	Decoder Quantization Scale information of Current Frame Starting Address	**MP4_DEC_QS_ADDR**
MP4 + 0280h	MPEG4 Decoder Cycle Count	**MP4_DEC_CYCLE_COUNT**
MP4 + 0280h	Core Configuration Register	**MP4_CORE_CONF**
MP4 + 0300h	Core Encoder Configuration Register	**MP4_CORE_ENC_CONF**
MP4 + 0304h	Core Duplex Controller Status Register	**MP4_DUPLEX_STS**
MP4 + 0314h	Current VOP Base Address Register	**MP4_CORE_VOP_ADDR**
MP4 + 0318h	Core Reference VOP Base Address Register	**MP4_CORE_REF_ADDR**
MP4 + 031Ch	Core Reconstructed VOP Base Address Register	**MP4_CORE_REC_ADDR**
MP4 + 0324h	Core VLE Data Load-Store LSB Base Address Register	**MP4_CORE_DATA_STORE_ADD R**
MP4 + 0328h	Core DC/AC Prediction Storage LSB Base Address Register	**MP4_CORE_DACP_ADDR**
MP4 + 032Ch	Core Motion Vector Storage LSB Base Address Register	**MP4_CORE_MVP_ADDR**
MP4 + 0330h	Core VOP Structure 0 Register	**MP4_CORE_VOP_STRUCT0**
MP4 + 0334h	Core VOP Structure 1 Register	**MP4_CORE_VOP_STRUCT1**

MP4 + 0338h	Core VOP Structure 2 Register	**MP4_CORE_VOP_STRUCT2**
MP4 + 033Ch	Core VOP Structure 3 Register	**MP4_CORE_VOP_STRUCT3**
MP4 + 0340h	Core MB Structure 0 Register	**MP4_CORE_MB_STRUCT0**
MP4 + 0344h	Core MB Structure 1 Register	**MP4_CORE_MB_STRUCT1**
MP4 + 0348h	Core MB Structure 2 Register	**MP4_CORE_MB_STRUCT2**
MP4 + 034Ch	Core MB Structure 3 Register	**MP4_CORE_MB_STRUCT3**
MP4 + 0350h	Core MB Structure 4 Register	**MP4_CORE_MB_STRUCT4**
MP4 + 0354h	Core MB Structure 5 Register	**MP4_CORE_MB_STRUCT5**
MP4 + 0358h	Core MB Structure 6 Register	**MP4_CORE_MB_STRUCT6**
MP4 + 035Ch	Core MB Structure 7 Register	**MP4_CORE_MB_STRUCT7**
MP4 + 0370h	Core VLC DMA Status Register	**MP4_CORE_VLC_DMA_STS**
MP4 + 0374h	Core VLE Status Register	**MP4_CORE_VLE_STS**
MP4 + 0378h	Core VLC DMA Base Address Register	**MP4_CORE_VLC_BASE_ADDR**
MP4 + 037Ch	Core VLC DMA Base Bit Count Register	**MP4_CORE_VLC_BASE_BITCNT**
MP4 + 0380h	Core VLC DMA Buffer Limit Register	**MP4_CORE_VLC_LIMIT**
MP4 + 0384h	Core VLC DMA Current Word Register	**MP4_CORE_VLC_WORD**
MP4 + 0388h	Core VLC DMA Current Bit Count Register	**MP4_CORE_VLC_BITCNT**
MP4 + 038Ch	Core VLC DMA Ring Buffer Ending Address Register	**MP4_CORE_VLC_JUMP_FROM_ADDR**
MP4 + 0390h	Core VLC DMA Ring Buffer Starting Address Register	**MP4_CORE_VLC_JUMP_TO_ADDR**

5.23.2.2 Main Control

MP4+0000h Video CODEC Command Register MP4_CODEC_COMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	DEC_START	ENC_START	DEC_RST	ENC_RST	CORE_RST
Type												WO	WO	WO	WO	WO

This register is the main command register for video CODEC.

CORE_RST Software reset control. Writing 1 to this bit will reset the hardware core excluding APB control register set of decoder and encoder. Only do the reset after 1 frame done or when codec is idle.

ENC_RST Software reset control. Writing 1 to this bit will reset the APB control register set of encoder. Note that this bit won't reset the hardware core. Only do the reset after 1 frame done or when codec is idle.

DEC_RST Software reset control. Writing 1 to this bit will reset the APB control register set of decoder. Note that this bit won't reset the hardware core. Only do the reset after 1 frame done or when codec is idle.

ENC_START Start the encode operation if writing 1 to this bit. The encode operation will start only when no decode operation is running; otherwise the encode operation will queue until decode operation is done.

DEC_START Start the decode operation if writing 1 to this bit. The decode operation will start only when no encode operation is running; otherwise the decode operation will queue until encode operation is done.

MP4+0004h VLC DMA Command Register MP4_VLC_DMA_CMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**RESUME**	**STOP**
Type															WO	WO

This register is the main control of VLC DMA.

STOP Stop the VLC DMA. Stop VLC DMA activities through SW rather than HW state machine.

RESUME Resume the VLC DMA access. VLC DMA state machine will go to a pending state if the maximum allowed write count to target memory is reached and then an interrupt has occurred. After re-allocating the target address, SW writes RESUME to unfreeze the encoding process.

5.23.2.3 Encoder

5.23.2.3.1 Control

MP4+0100h Encoder Configuration Register MP4_ENC_CODEC_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**PREFETCH**	**CHECK_TV**
Type															R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	**PMV**	**DQUAN**	-	**HALF**	**STEP_LIMIT**				**VPGOB**	**DCT**	**IRQ**	**ENC**
Type					R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the operating conditions and modes of video CODEC.

ENC Video CODEC Operation Mode
- 0 Decode Mode
- 1 Encode Mode

IRQ Control for interrupt request
- 0 Disable the interrupt reporting mechanism
- 2 Enable the interrupt reporting mechanism

DCT DCT Control
- 0 Enable JPEG CODEC Operation

2 Enable MPEG-4 Video CODEC Operation

VPGOB Control for decoding Video Packet Header. (No use in codec encoding mode)

0 Disable: decoding in Video Packet Level. It means the software will take the responsibility for decoding packet header of each video packet.

1 Enable: decoding in Video Object Plane Level

STEP_LIMIT Step limit for Motion Estimation. The total number of steps in a n-step search is STEP_LIMIT+2. Increasing STEP_LIMIT can increase search range of motion vectors.

HALF Motion Estimation uses half-pel resolution

0 Disable. Perform full pel motion estimation only

1 Enable. Perform full pel motion estimation first, then half pel motion estimation

DQUAN Control for automatic update quantizer_scale process. (No use in codec encoding mode)

0 Disable

1 Enable

PMV Predictive Motion Vector Search. This is a two-path search algorithm. The idea is to initially consider several highly likely predictors (starting points), perform motion estimation from these predictors, and choose the best result among these predictors. In our approach, the two predictors approach is adopted. The origin (0,0) is considered as the predictor of first path. The minimum BDM point found in first pass will be the predictor of the second path. After finishing two-path motion estimation, choose the best result between the two minimum BDM points. This algorithm can significantly improve PSNR by about 0.8dB. However, the search time will increase by about 60%. Setting PMV to 1 or 0 is the trade-off between visual quality and search time.

0 Disable

1 Enable

CHECK_TV Enable signal to check if TV codec is busy before starting encoding operation.

0 Do not check TV codec

1 Check TV codec

PREFETCH In order to reduce external memory access times, fetch reference frame and current frame from external memory into internal memory before encoding. Pre-fetch do not support frames which width less than 128-pixel or height less than 96-pixel.

0 Disable

1 Enable

MP4+0104h Encoder Status Register MP4_ENC_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STATE															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STATE															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register provides the state information of encoding sequencer for software program. It is a mirror of the HW one-hot sequencer state machine and can be used for debugging or IRQ status judging.

MP4+0108h Encoder Interrupt Mask Register MP4_ENC_IRQ_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	**DMA**	**PACK**		**ENC_DONE**
Type													R/W	R/W		R/W
Reset													1	1		1

This register contains mask bit for each interruption source in MPEG-4 Video encoder. It allows each interrupt source to be disabled or masked out separately under software control. After System Reset or software reset, all bit values will be set to '0' to indicate that interrupt requests are enabled.

ENC_DONE Mask of encode complete interruption.

PACK Mask of video packet bit count expire interruption.

DMA Mask of VLE DMA Limit interruption.

MP4+010Ch Encoder Interrupt Status Register MP4_ENC_IRQ_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	-	**DMA**	**PACK**		**ENC_DONE**
Type													RO	RO		RO

This register allows software program to poll which interrupt source generates the interrupt request. A bit set to '1' indicates a corresponding active interrupt source. Note that IRQ control bit in MP4_ENC_CODEC_CONF should be enabled first in order to activate the interrupt reporting mechanism.

ENC_DONE Encode complete. A normal condition when encoding procedure is done.

PACK Video Packet Bit Count Exceed interrupt. If a video packet size is larger than defined the interrupt will happen.

DMA Mask of VLC DMA interruption. When decoder detects empty VLD stream buffer, an interrupt will inform the driver SW to refill the VLD stream buffer.

MP4+0110h Encoder Interrupt Acknowledge Register MP4_ENC_IRQ_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	-	-	-	-	-	-	-	-	-	-	-	-	**DMA**	**PACK**		**ENC_ DONE**
Type													WC	WC		WC

This register provides a mean for software program to acknowledge the interrupt source. Writing a '1' to the specific bit position will result in an acknowledgement to the corresponding interrupt source and clear the corresponding bit in MP4_ENC_IRQ_STS.

ENC_DONE Encode Task Complete.

PACK VIDEO Packet Bit Count Expired

DMA VLC DMA Buffer Limit Reached.

MP4+0114h Encoder Configuration Register MP4_ENC_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	**PACKCNT**										**PACK**
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	**INTRA**						-	-	**SKIP**					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0			0	0	0	0	0	0

This register is used specially to configure the desired encode conditions and modes for video CODEC.

SKIP Threshold for deciding not_coded bit. The value of SKIP is programmed by software first. The first round of pattern code (*me_pattern_code* is set to 6'h0 whenever (SAD_y + SDA_u + SAD_v) <= skip_threshold*16 *not_coded* bit will be set if *pattern_code* = 6'h0 and motion vector = (0,0)

INTRA Threshold for deciding INTRA Coding in P frame. The value of INTRA is programmed by software first. The 3-bits macro-block type (*mb_type*) is set to 3'h0 (Inter MB) if SADy < *intra_threshold**1024. Otherwise, *mb_type* is set to 3'h3 (Intra_MB)

PACK Use Video Packet Mode

- 0 Disable
- 1 Enable

PACKCNT Desired Bit Counts for a Video Packet. Used in encode mode to define the largest VLE buffer size of a video packet.

5.23.2.3.2 Base Addresses

MP4+0124h Encoder Current VOP Base Address Register MP4_ENC_VOP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**VOP**															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VOP**													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

This register describes the starting address of Current VOP Frame that is going to be encoded. Note that this base address should be 8-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

VOP Current VOP Base Address

MP4+0128h Encoder Reference VOP Base Address Register MP4_ENC_REF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REF															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REF													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

This register describes the starting address of Reference VOP Frame. Note that this base address should be 8-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REF Reference VOP Base Address

MP4+012Ch Encoder Reconstructed VOP Base Address Register MP4_ENC_REC_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REC															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REC													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

This register describes the starting address of Reconstructed VOP Frame. Note that this base address should be 8-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REC Reconstructed VOP Base Address

MP4+0130h VLE Data Load-Store LSB Base Address Register MP4_ENC_DATA_STORE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STORE															
Type															R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STORE														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of VLE Data Load-Store buffer in data-partitioned mode. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be: 3K bytes and (number of macroblock per frame * 32) bytes, respectively.

STORE LSB address of VLE Data Load-Store buffer

MP4+0134h DC/AC Prediction Storage LSB Base Address Register MP4_ENC_DACP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DACP															
Type															R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DACP														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of DC/AC Prediction Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be 2K bytes and 4K bytes, respectively.

DACP LSB address of DC/AC Prediction Storage buffer

MP4+0138h Motion Vector Storage LSB Base Address Register MP4_ENC_MVP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MVP_ADDR															
Type															R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MVP_ADDR														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of Motion Vector Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder should be mb_x_limit * 2 * 4 bytes, which equals to 320 Bytes for VGA size.

MVP_ADDR LSB address of Motion Vector Storage buffer

5.23.2.3.3 Data Structure

MP4+0140h Encoder VOP Structure 0 Register MP4_ENC_VOP_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-		ROUND
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLCTHR			QUANT					FCODE			SHORT	-	RVLC	DATA	TYPE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W

This register is used to describe the header information of a certain Video Object Plane that is going to be processed by video CODEC.

TYPE vop_coding_type definition, for both decode and encode.

- 0 This is a P-VOP frame (inter frame)
- 1 This is an I-VOP frame (intra frame)

DATA data_partitioned for decode only. (No use in codec encoding mode)

0 Data stream is in non-data-partitioned mode

1 Data stream is in data-partitioned mode

RVLC resversible_vlc, for decode only. (No use in codec encoding mode)

0 Data stream contains no reversible VLC information

1 Data stream uses reversible VLC tables.

SHORT short_video_header; for both decode and encode

0 Normal MPEG-4 format

1 H.263 Compatible format

FCODE fcode setting for both decode and encode, ranges from 0 to 7.

QUANT vop_quant for both decode and encode. Quantizer scale of the current frame. For variable Q in decode mode, QUANT is an initial setting of the current frame.

VLCTHR intra_dc_vlc_thr for decode only. According to VLCTHR, the decoder has to switch from intra DC mode to inter DC mode when the quantizer_scale is larger than a pre-defined value. VLCTHR ranges from 0 to 7.

ROUND Rounding type of half-pel motion compensation. ROUND==1 means truncation toward zero (the pixel value is always larger than 0); ROUND==0 means rounding-off addition.

ROUND Rounding type of half-pel motion compensation. ROUND==1 means truncation toward zero (the pixel value is always larger than 0); ROUND==0 means rounding-off addition.

MP4_ENC_VOP_STRUCT1 Encoder VOP Structure 1 Register

MP4+0144h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	HECBIT					-	-	-	-	MBLENGTH			
Type				R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	YLIMIT					-	-	-	XLIMIT				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

XLIMIT Macroblock count in X direction of a frame.

YLIMIT Macroblock count in Y direction of a frame.

MBLENGTH Bit count of Macroblock Number in Video Packet Header. It is a value defined by the following formula: MBCNT = XLIMIT * YLIMIT. For larger MBCNT, we have larger MBLENGTH. MBLENGTH is ranged from 1 to 14.

HECBIT Bit count of extension header code in Video Packet Header

MP4+0148h Encoder VOP Structure 2 Register MP4_ENC_VOP_STRUCT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	**MBNO**								
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	**VP_YPOS**					-	-	-	**VP_XPOS**				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

VP_XPOS Starting position of the current Video Packet in X coordinate.

VP_YPOS Starting position of the current Video Packet in Y coordinate.

MBNO Macroblock count limit for a video packet or frame. For a CIF frame the value will be 396.

MP4+014Ch VOP Structure 3 Register MP4_VOP_STRUCT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	**MBNO**								
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	**YPOS**					-	-	-	**XPOS**				
Type				RO	RO	RO	RO	RO				RO	RO	RO	RO	RO

This register provides the position and count information of a certain macroblock that is currently under process of video CODEC.

XPOS Current Macroblock Position in X coordinate

YPOS Current Macroblock Position in Y coordinate

MBNO Current Macroblock Count

MP4+0150h MB Structure 0 Register MP4_ENC_MB_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	**QUANTIZER**		
Type														R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**QUANTIZER**		**DCVLC**	**AC**	**DQUANT**		**PATTERN**						**TYPE**			**CODED**
Type	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the header information of current macroblock. This register is mostly used for debugging. Also used to provide hardware certain header information if all header parsing is done by software instead of hardware.

CODED not_coded flag of current macroblock; not_coded can be decoded by hardware from macroblock header.

TYPE mb_coding_type of current macroblock; mb_coding_type can be decoded by hardware from mcbpc in macroblck header.

PATTERN pattern_code of current macroblock; pattern_code can be decoded by hardware from cbpc and cbpy in macroblock header.

DQUANT dquant. It can be –2, -1, +1 or +2; total 4 possible choices using 2 bits to represent; dquant can be decoded by hardware from macroblock header.

AC ac_pred_flag. It decides whether AC prediction is needed; always 0 in encoder; ac_pred_flag can be decoded by hardware from macroblock header.

DCVLC use_intra_dc_vlc. If this bit is 0, intra AC VLC decode is used (no intra DC exists in current macroblock).

QUANTIZER quantizer_scale, ranged from 1 to 31. It can be variable if we have dquant values.

5.23.2.3.4 VLC DMA

MP4+0160h Encoder VLC DMA Base Address Register MP4_ENC_VLC_BASE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASE														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register is used to describe the address of started Code Word for each VLC DMA buffer. Note that this base address should be 4-byte aligned.

BASE VLC DMA Base Address

MP4+0164h Encoder VLC DMA Base Bit Count Register MP4_ENC_VLC_BASE_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	BIT				
Type												R/W	R/W	R/W	R/W	R/W

This register is used to describe the starting bit position of the 1st Code Word in the 1st VLC DMA buffer. For the following VLC DMA buffers, it is assumed that they are all 4-byte aligned and always start from bit position "0".

BIT Start of Bit at the 1st Code Word of 1st DMA Buffer

MP4+0168h Encoder VLC DMA Buffer Limit Register MP4_ENC_VLC_LIMIT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LIMIT**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register is used to describe the buffer size of each VLC DMA buffer. Note that the value is counted in 4 words (4*32-bit). Whenever the limit is reached and the corresponding interrupt control is enabled, an interrupt request will be generated.

LIMIT DMA Buffer Size, Count in Word (32-bit)

LIMIT DMA Buffer Size, Count in Word (32-bit)

MP4+016Ch Encoder VLC DMA Current Word Register MP4_ENC_VLC_WORD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register provides the address information of a certain code word that is under process of video CODEC. SW reads it back after encode of a frame is done.

ADDR VLC DMA current Address

MP4+0170h Encoder VLC DMA Current Bit Count Register MP4_ENC_VLC_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	**BITCNT**				
Type												RO	RO	RO	RO	RO

This register provides the bit position information of a certain Code Word that is under process of video CODEC.

BITCNT Current Bit Count

MP4+0174h Encoder VLC DMA Ring Buffer Ending Address Register MP4_ENC_VLC_JUMP_FROM_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_FROM_ADDR The ending address of the current DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR. To disable the ring buffer feature, set this register to all ones; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

MP4+0178h Encoder VLC DMA Ring Buffer Starting Address Register MP4_ENC_VLC_JUMP_TO_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_TO_ADDR The starting address of the next DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

5.23.2.3.5 Resync Marker

MP4+0180h MPEG4 Encoder Resync Marker Configuration 0 Register MP4_ENC_RESYNC_CONF0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EN	MODE	PERIOD_BITS													
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERIOD_BITS															

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

EN Resync marker insertion enable
- 0 Disable resync marker insertion
- 1 Enable resync marker insertion

MODE Resync Marker insertion mode selection
- 0 resync marker is inserted based on number of bits
- 1 resync marker is inserted based on number of macroblocks

PERIOD_BITS Period in number of bits to insert resync marker; only effective when MODE is set to 0; hardware will insert resync marker at the next macroblock boundary once the bit length of a video packet exceeds this value.

MP4+0184h MPEG4 Encoder Resync Marker Configuration 1 Register MP4_ENC_RESYNC_CONF1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**HEC**
Type																R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PERIOD_MB**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

HEC Header Extension Code; indicates the value of header_extension_code in MPEG4 standard (ISO/IEC 14496-2)
- 0 header_extension_code is 0.
- 1 header_extension_code is 1.

PERIOD_MB Period in number of macroblocks (MB) to insert resync marker; only effective when MODE is set to 1; hardware will insert resync marker at the next macroblock boundary once the number of macroblock in current video packet exceeds this value.

MP4+0188h MPEG4 Encoder Local Time Base Register MP4_ENC_TIME_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	**MODULO_TIME_BASE**					**BW**			
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VOP_TIME_INCREMENT**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MODULO_TIME_BASE Represent the value of modulo_time_base; value ranges from 0 to 31.

BW Bit width of vop_time_increment. The real bit-width of vop_time_increment is (BW + 1), ranging from 1 to 16.

VOP_TIME_INCREMENT Carries the value of vop_time_increment defined in MPEG4 standard (ISO/IEC 14496-2); the meaningful bit width of vop_time_increment is signaled by BW field.

MP4+018Ch MPEG4 Encoder Pre-fetch Mode Internal REF Base Address MP4_ENC_REF_INT_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MP4_ENC_REF_INT_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MP4_ENC_REF_INT_ADDR													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MP4_ENC_REF_INT_ADDR Represent the base address of reference frame in internal memory. It is 8-byte aligned, and need search range size. Only work in encoder pre-fetch mode. Fetch reference frame data to internal memory.

Minimum internal memory size = frame_width * (56+32) = frame_width * 88.

MP4_ENC_REF_INT_ADDR can’t be the beginning address of internal memory, it must have 32 bytes shift.

And also, the end of reference internal memory must have 16-byte room to the last address of internal memory.

MP4+0190h MPEG4 Encoder Pre-fetch Mode Internal CUR Base Address MP4_ENC_CUR_INT_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MP4_ENC_CUR_INT_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MP4_ENC_CUR_INT_ADDR													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MP4_ENC_CUR_INT_ADDR Represent the base address of current frame in internal memory. It is 8-byte aligned, and need 3-macroblock size. Only work in encoder pre-fetch mode. Fetch current frame data to internal memory.

MP4+0194h MPEG4 Encoder Cycle Count MP4_ENC_CYCLE_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CYCLE_COUNT															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CYCLE_COUNT															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register is used to store the total cycle count of encoder in current frame from start to end.

5.23.2.4 Decoder

MP4+0200h Decoder Configuration Register MP4_DEC_CODEC_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	PMV	DQUAN	-	HALF	STEP_LIMIT				VPGOB	DCT	IRQ	ENC
Type					R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the operating conditions and modes of video CODEC.

ENC Video CODEC Operation Mode
- 0 Decode Mode
- 1 Encode Mode

IRQ Control for interrupt request
- 0 Disable the interrupt reporting mechanism
- 1 Enable the interrupt reporting mechanism

DCT DCT Control
- 0 Enable JPEG CODEC Operation
- 1 Enable MPEG-4 Video CODEC Operation

VPGOB Control for decoding Video Packet Header.
- 0 Disable: decoding in Video Packet Level. It means the software will take the responsibility for decoding packet header of each video packet.
- 1 Enable: decoding in Video Object Plane Level

STEP_LIMIT Step limit for Motion Estimation, for encode only; keep this for legacy reason. The total number of steps in a n-step search is STEP_LIMIT+2. Increasing STEP_LIMIT can increase search range of motion vectors.

HALF Motion Estimation uses half-pel resolution, for encode only. (No use in codec decoding mode)
- 0 Disable. Perform full pel motion estimation only
- 1 Enable. Perform full pel motion estimation first, then half pel motion estimation

DQUAN Control for automatic update quantizer_scale process.
- 0 Disable
- 1 Enable

PMV Predictive Motion Vector Search, for encode only; keep this for legacy reason. This is a two pass search algorithm. This algorithm can co-operate with both four step search (FME=0) and Mediatek proprietary search (FME=1). The idea is to initially consider several highly likely predictors (starting points), perform motion estimation from these predictors, and choose the best result among these predictors. In our approach, the two predictors approach is adopted. The origin (0,0) is considered as the predictor of first pass. The minimum BDM point found in first pass will be the predictor of the second pass. After finishing two-pass motion estimation, choose the best result between the two minimum BDM points. This algorithm can significantly improve PSNR by about 0.8dB. However, the search time will increase by about 60%. Setting PMV to 1 or 0 is the trade-off between visual quality and search time. (No use in codec decoding mode)

0 Disable
1 Enable

MP4+0204h Decoder Status Register MP4_DEC_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STATE															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STATE															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register provides the state information of decoding sequencer for software program. It is a mirror of the HW one-hot sequencer state machine and can be used for debugging or IRQ status judging.

MP4+0208h Decoder Interrupt Mask Register MP4_DEC_IRQ_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	MB_MISMATCH_ERR	DMA		DEC_DONE	MARKER	RLD	VLD
Type										R/W	R/W		R/W	R/W	R/W	R/W
Reset										1	1	1	1	1	1	1

This register contains mask bit for each interrupt sources in MPEG-4 Video Decoder. It allows each interrupt source to be disabled or masked out separately under software control. After System Reset or software reset, all bit values will be set to ‘0’ to indicate that interrupt requests are enabled.

MB_MISMATCH_ERR Mask of MB_MISMATCH_ERR interrupt.

DMA Mask of VLC DMA interrupt.

DEC_DONE Mask of decode complete interrupt.

MARK Mask of marker error interrupt in decode.

RLD Mask of run length coding error interrupt

VLD Mask of VLD error interrupt generated in decoding process.

MP4+020Ch Decoder Interrupt Status Register MP4_DEC_IRQ_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	-	-	-	-	-	-	-	-	-	MB_MISMATCH_ERR	DMA		DEC_DONE	MARK	RLD	VLD
Type										RO	RO		RO	RO	RO	RO

This register allows software program to poll which interrupt source generates the interrupt request. A bit set to '1' indicates a corresponding active interrupt source. Note that IRQ control bit in MP4_DEC_CODEC_CONF should be enabled first in order to activate the interrupt reporting mechanism.

MB_MISMATCH_ERR MB_MISMATCH_ERR interrupt, when macroblock_number error occurs, and the error number exceeds the limitation of total macroblock counts.

DMA Mask of VLC DMA interrupt. When decoder detects empty VLD stream buffer, an interrupt will inform the driver SW to refill the VLD stream buffer.

DEC_DONE Decode complete. A normal condition when decoding procedure is done.

MARK Marker decode error occurred.

RLD Run length coding error. Generated when the accumulated run value is larger than 64 (the 8x8 block memory size).

VLD VLD error of decoding process. Generated when a code can not be correctly referenced in VLD table

MP4+0210h Decoder Interrupt Acknowledge Register MP4_DEC_IRQ_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	MB_MISMATCH_ERR -	DMA		DEC_DONE	MARK	RLD	VLD
Type										WC	WC		WC	WC	WC	WC

This register provides a mean for software program to acknowledge the interrupt source. Writing a '1' to the specific bit position will result in an acknowledgement to the corresponding interrupt source.

VLD Variable Length Decoding Error
RLD Run Length Decoding Error
MARK Marker Decoding Error
DEC_DONE Decode Task Complete
DMA VLC DMA Buffer Limit Reached
MB_MISMATCH_ERR MB_MISMATCH_ERR interrupt, when macroblock_number error occurs, and the error number exceeds the limitation of total macroblock counts.

5.23.2.4.1 Base Address

MP4+0224h Decoder Reference VOP Base Address Register MP4_DEC_REF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REF_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REF_ADDR														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

This register describes the starting address of Reference VOP Frame. Note that this base address should be 8-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REF Reference VOP Base Address.

MP4+0228h Decoder Reconstructed VOP Base Address Register MP4_DEC_REC_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REC_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REC_ADDR													-	-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

This register describes the starting address of Reconstructed VOP Frame. Note that this base address should be 8-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REC Reconstructed VOP Base Address.

MP4+0230h Decoder Data Load-Store LSB Base Address Register MP4_DEC_DATA_STORE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STORE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STORE														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of memory buffer used to store the macroblock header, Intra DC values and motion vectors as decoding data-partitioned MPEG4 files. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be: 3K bytes and (number of macroblock per frame * 32) bytes, respectively.

STORE LSB address of VLE Data Load-Store buffer

MP4+0234h DC/AC Prediction Storage LSB Base Address Register MP4_DEC_DACP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DACP															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DACP														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of DC/AC Prediction Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be: 2K bytes and 4K bytes, respectively.

DACP LSB address of DC/AC Prediction Storage buffer

MP4+0238h Motion Vector Storage LSB Base Address Register MP4_DEC_MVP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MVP_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MVP_ADDR														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of Motion Vector Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder should be mb_x_limit * 2 * 4 bytes, which equals to 320 Bytes for VGA size.

MVP_ADDR LSB address of Motion Vector Storage buffer

5.23.2.4.2 Data Structure

MP4+0240h Decoder VOP Structure 0 Register MP4_DEC_VOP_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ROUND
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLCTHR			QUANT					FCODE			SHORT	-	RVLC	DATA	TYPE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W

This register is used to describe the header information of a certain Video Object Plane that is going to be processed by video CODEC.

TYPE vop_coding_type definition, for both decode and encode.

- 0 This is a P-VOP frame (inter frame)
- 1 This is an I-VOP frame (intra frame)

DATA data_partitioned, for decode only.

- 0 Data stream is in non-data-partitioned mode

1 Data stream is in data-partitioned mode

RVLC resversible_vlc, for decode only.

0 Data stream contains no reversible VLC information

1 Data stream uses reversible VLC tables.

SHORT short_video_header; for both decode and encode

0 Normal MPEG-4 format

1 H.263 Compatible format

FCODE fcode size setting for both decode and encode, ranges from 0 to 7.

QUANT vop_quant. For both decode and encode. Quantizer scale of the current frame. For variable Q in decode mode, QUANT is an initial setting of the current frame.

VLCTHR intra_dc_vlc_thr. For decode only. According to VLCTHR, the decoder has to switch from intra DC mode to inter DC mode when the quantizer_scale is larger than a pre-defined value. VLCTHR ranges from 0 to 7.

ROUND Rounding type of half-pel motion compensation. ROUND==1 means truncation toward zero (the pixel value is always larger than 0); ROUND==0 means rounding-off addition.

MP4+0244h Decoder VOP Structure 1 Register MP4_DEC_VOP_STRUCT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	HECBIT					-	-	-	-	MBLENGTH			
Type				R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	YLIMIT					-	-	-	XLIMIT				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

XLIMIT Macroblock count in X direction of a frame.

YLIMIT Macroblock count in Y direction of a frame.

MBLENGTH Bit count of Macroblock Number in Video Packet Header. It is a value defined by the following formula: MBCNT = XLIMIT * YLIMIT. For larger MBCNT, we have larger MBLENGTH. MBLENGTH is ranged from 1 to 14.

HECBIT Bit count of header extension code in Video Packet Header; this section includes modulo_time_base, vop_time_increment, vop_coding_type, intra_dc_vlc_thr and vop_fcode_forward(only in P-VOP).

MP4+0248h Decoder VOP Structure 2 Register MP4_DEC_VOP_STRUCT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	MBNO								
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	VP_YPOS					-	-	-	VP_XPOS				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

VP_XPOS Starting position of the current Video Packet in X coordinate.
VP_YPOS Starting position of the current Video Packet in Y coordinate.
MBNO Macroblock count limit for a video packet or frame. For a CIF frame the value will be 396.

MP4+024Ch Decoder MB Structure 0 Register MP4_DEC_MB_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	QUANTIZER		
Type														R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	QUANTIZER		DCVLC	AC	DQUANT		PATTERN						TYPE			CODED
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the header information of the current macroblock. This register is mostly used for debugging. Also used to provide hardware certain header information if all header parsing is done by software instead of hardware.

CODED not_coded flag of current macroblock; not_coded can be decoded by hardware from macroblock header.
TYPE mb_coding_type of current macroblock; mb_coding_type can be decoded by hardware from mcbpc in macroblck header.
PATTERN pattern_code of current macroblock; pattern_code can be decoded by hardware from cbpc and cbpy in macroblock header.
DQUANT dquant. It can be –2, -1, +1 or +2; total 4 possible choices using 2 bits to represent; dquant can be decoded by hardware from macroblock header.
AC ac_pred_flag. It decides whether AC prediction is needed; always 0 in encoder; ac_pred_flag can be decoded by hardware from macroblock header.
DCVLC use_intra_dc_vlc. If this bit is 0, intra AC VLC decode is used (no intra DC exists in current macroblock).
QUANTIZER quantizer_scale, ranged from 1 to 31. It can be variable if we have dquant values.

5.23.2.4.3 VLC DMA

.MP4+0260h Decoder VLC DMA Base Address Register MP4_DEC_VLC_BASE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASE															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASE														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register is used to describe the address of started Code Word for each VLC DMA buffer. Note that this base address should be 4-byte aligned.

BASE VLC DMA Base Address

MP4+0264h Decoder VLC DMA Base Bit Count Register MP4_DEC_VLC_BASE_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	**BIT**				
Type												R/W	R/W	R/W	R/W	R/W

This register is used to describe the starting bit position of the 1st Code Word in the 1st VLC DMA buffer. For the following VLC DMA buffers, it is assumed that they are all 4-byte aligned and always start from bit position "0".

BIT Start of Bit at the 1st Code Word of 1st DMA Buffer

MP4+0268h Decoder VLC DMA Buffer Limit Register MP4_DEC_VLC_LIMIT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LIMIT**															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register is used to describe the buffer size of each VLC DMA buffer. Note that the value is counted in word (32-bit). Whenever the limit is reached and the corresponding interrupt control is enabled, an interrupt request will be generated.

LIMIT DMA Buffer Size, Count in Word (32-bit)

MP4+026Ch Decoder VLC DMA Current Word Register MP4_DEC_VLC_WORD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ADDR**															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADDR**															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register provides the address information of a certain code word that is under process of video CODEC. SW reads it back after decode of a frame is done.

ADDR VLC DMA current Address

MP4+0270h Decoder VLC DMA Current Bit Count Register MP4_DEC_VLC_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	-	-	-	-	-	-	-	-	-	-	-	BITCNT				
Type												RO	RO	RO	RO	RO

This register provides the bit position information of a certain Code Word that is under process of video CODEC.

BITCNT Current Bit Count

MP4+0274h Decoder VLC DMA Ring Buffer Ending Address Register MP4_DEC_VLC_JUMP_FROM_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_FROM_ADDR The ending address of the current DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR. To disable the ring buffer feature, set this register to all ones; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

MP4+0278h Decoder VLC DMA Ring Buffer Starting Address Register MP4_DEC_VLC_JUMP_TO_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_TO_ADDR The starting address of the next DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

MP4+027Ch Decoder Quantization Scale information of Current Frame Starting Address MP4_DEC_QS_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	QS_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	QS_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

QS_ADDR The starting address of the quantization scale and not_coded information in current frame for deblocking filter. The buffer size for decoder is: (((mb_x_limit+3)/4) * 4 * mb_y_limit) bytes.

MP4+0280h MPEG4 Decoder Cycle Count MP4_DEC_CYCLE_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CYCLE_COUNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CYCLE_COUNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is used to store the total cycle count of decoder in current frame from start to end.

5.23.2.5 Core

MP4+0300h Core Configuration Register MP4_CORE_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PREFETCH	-
Type															R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	PMV	DQUAN	-	HALF	STEP_LIMIT				VPGOB	DCT	IRQ	ENC
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the operating conditions and modes of video CODEC.

ENC Video CODEC Operation Mode
- 0 Decode Mode
- 1 Encode Mode

IRQ Control for interrupt request
- 0 Disable the interrupt reporting mechanism
- 1 Enable the interrupt reporting mechanism

DCT DCT Control
- 0 Enable JPEG CODEC Operation
- 1 Enable MPEG-4 CODEC Operation

VPGOB Control for decoding Video Packet Header.
- 0 Disable: decoding in Video Packet Level. It means the software will take the responsibility for decoding packet header of each video packet.

1 Enable: decoding in Video Object Plane Level

STEP_LIMIT Step limit for Motion Estimation. The total number of steps in a n-step search is STEP_LIMIT+2. Increasing STEP_LIMIT can increase search range of motion vectors.

HALF Motion Estimation uses half-pel resolution

- 0 Disable. Perform full pel motion estimation only
- 1 Enable. Perform full pel motion estimation first, then half pel motion estimation

DQUAN Control for automatic update quantizer_scale process.

- 0 Disable
- 1 Enable

PMV Predictive Motion Vector Search. This is a two pass search algorithm. This algorithm can co-operate with both four step search (FME=0) and Mediatek proprietary search (FME=1). The idea is to initially consider several highly likely predictors (starting points), perform motion estimation from these predictors, and choose the best result among these predictors. In our approach, the two predictors approach is adopted. The origin (0,0) is considered as the predictor of first pass. The minimum BDM point found in first pass will be the predictor of the second pass. After finishing two-pass motion estimation, choose the best result between the two minimum BDM points. This algorithm can significantly improve PSNR by about 0.8dB. However, the search time will increase by about 60%. Setting PMV to 1 or 0 is the trade-off between visual quality and search time.

- 0 Disable
- 1 Enable

MP4+0304h Core Encoder Configuration Register MP4_CORE_ENC_CONF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	PACKCNT										PACK
Type						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset						0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	INTRA						-	-	SKIP					
Type			R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0			0	0	0	0	0	0

This register is used specially to configure the desired encode conditions and modes for video CODEC.

SKIP Threshold for deciding not_coded bit. The value of SKIP is programmed by software first. The first round of pattern code (*me_pattern_code* is set to 6’h0 whenever (SAD_y + SDA_u + SAD_v) <= skip_threshold*16 *not_coded* bit will be set if *pattern_code* = 6’h0 and motion vector = (0,0)

INTRA Threshold for deciding INTRA Coding in P frame. The value of INTRA is programmed by software first. The 3-bits macro-block type (*mb_type*) is set to 3’h0 (Inter MB) if SADy < *intra_threshold**1024. Otherwise, *mb_type* is set to 3’h3 (Intra_MB)

PACK Use Video Packet Mode

- 0 Disable
- 1 Enable

PACKCNT Desired Bit Counts for a Video Packet. Used in encode mode to define the largest VLE buffer size of a video packet

MP4+0308h Core Duplex Controller Status Register MP4_DUPLEX_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	DUPLEX_STATE						
Type										RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DUPLEX_STATE															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register is used to read back the current state of duplex controller in MPEG4 Codec.

DUPLEX_STATE Current state of duplex controller.

5.23.2.5.1 Base Addresses

CODEC MPEG-4/H.263 CODEC MSB Base Address

MP4+0314h Current VOP Base Address Register MP4_CORE_VOP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VOP															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VOP														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the starting address of Current VOP Frame that is going to be encoded. Note that this base address should be 4-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

VOP Current VOP Base Address.

MP4+0318h Core Reference VOP Base Address Register MP4_CORE_REF_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REF															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REF														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the starting address of Reference VOP Frame. Note that this base address should be 4-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REF Reference VOP Base Address.

MP4+031Ch Core Reconstructed VOP Base Address Register MP4_CORE_REC_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REC															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REC														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the starting address of Reconstructed VOP Frame. Note that this base address should be 4-byte aligned. And the required frame buffer size should be: numbers of pixel per frame * 1.5 bytes (YUV420 format).

REC Reconstructed VOP Base Address.

MP4+0324h Core VLE Data Load-Store LSB Base Address Register MP4_CORE_DATA_STORE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STORE															
Type															R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	STORE														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of VLE Data Load-Store buffer in data-partitioned mode. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be: 3K bytes and (number of macroblock per frame * 32) bytes, respectively.

STORE LSB address of VLE Data Load-Store buffer

MP4+0328h Core DC/AC Prediction Storage LSB Base Address Register MP4_CORE_DACP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DACP															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DACP														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of DC/AC Prediction Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder and decoder should be: 2K bytes and 4K bytes, respectively.

DACP LSB address of DC/AC Prediction Storage buffer

MP4+032Ch Core Motion Vector Storage LSB Base Address Register MP4_CORE_MVP_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MVD_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MVD_ADDR														-	-
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

This register describes the LSB address of Motion Vector Storage buffer. Note that this base address should be 4-byte aligned. And the required buffer size for encoder should be mb_x_limit * 2 * 4 bytes, which equals to 320 Bytes for VGA size.

MVD_ADDR LSB address of Motion Vector Storage buffer

5.23.2.5.2 Data Structure

MP4+0330h Core VOP Structure 0 Register MP4_CORE_VOP_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ROUND
Type																R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLCTHR			QUANT					FCODE			SHORT	-	RVLC	DATA	TYPE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W

This register is used to describe the header information of a certain Video Object Plane that is going to be processed by video CODEC.

TYPE vop_coding_type definition, for both decode and encode.
- 0 This is a P-VOP frame (inter frame)
- 1 This is an I-VOP frame (intra frame)

DATA data_partitioned, for decode only.
- 0 Data stream is in non-data-partitioned mode
- 1 Data stream is in data-partitioned mode

RVLC resversible_vlc, for decode only.
- 0 Data stream contains no reversible VLC information
- 1 Data stream uses reversible VLC tables.

SHORT short_video_header; for both decode and encode
- 0 Normal MPEG-4 format
- 1 H.263 Compatible format

FCODE fcode size setting for both decode and encode, ranges from 0 to 7.

QUANT vop_quant. For both decode and encode. Quantizer scale of the current frame. For variable Q in decode mode, QUANT is an initial setting of the current frame.

VLCTHR intra_dc_vlc_thr. For decode only. According to VLCTHR, the decoder has to switch from intra DC mode to inter DC mode when the quantizer_scale is larger than a pre-defined value. VLCTHR ranges from 0 to 7.

ROUND Rounding type of half-pel motion compensation. ROUND==1 means truncation toward zero (the pixel value is always larger than 0); ROUND==0 means rounding-off addition.

MP4+0334h Core VOP Structure 1 Register MP4_CORE_VOP_STRUCT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	HECBIT					-	-	-	-	MBLENGTH			
Type				R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	YLIMIT					-	-	-	XLIMIT				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

 © 2009 MediaTek Inc.

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

XLIMIT Macroblock count in X direction of a frame.

YLIMIT Macroblock count in Y direction of a frame.

MBLENGTH Bit count of Macroblock Number in Video Packet Header. It is a value defined by the following formula: MBCNT = (XLIMIT+15)/16 * (YLIMIT+15)/16. For larger MBCNT, we have larger MBLENGTH. MBLENGTH is ranged from 1 to 14.

HECBIT Bit count of extension header code in Video Packet Header

MP4+0338h Core VOP Structure 2 Register MP4_CORE_VOP_STRUCT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	MBNO								
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	VP_YPOS					-	-	-	VP_XPOS				
Type				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W

This register is used by software program to control the start position and count limit of macroblock for a certain Video Packet or Video Object Plane that is going to be processed by video CODEC.

VP_XPOS Starting position of current Video Packet in X coordinate that the SW wants to update.

VP_YPOS Starting position of current Video Packet in Y coordinate that the SW wants to update.

MBNO Macroblock count limit for a video packet or frame. For a CIF frame the value will be 396.

MP4+033Ch Core VOP Structure 3 Register MP4_CORE_VOP_STRUCT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	MBNO								
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	YPOS					-	-	-	XPOS				
Type				RO	RO	RO	RO	RO				RO	RO	RO	RO	RO

This register provides the position and count information of a certain macroblock that is currently under process of video CODEC.

XPOS Current Macroblock Position in X coordinate

YPOS Current Macroblock Position in Y coordinate

MBNO Current Macroblock Count

MP4+0340h Core MB Structure 0 Register MP4_CORE_MB_STRUCT0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	QUANTIZER		
Type														R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	QUANTIZER		DCVLC	AC	DQUANT		PATTERN						TYPE			CODED
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the header information of current macroblock. This register is mostly used for debugging. Also used to provide hardware certain header information if all header parsing is done by software instead of hardware.

CODED not_coded flag of current macroblock; not_coded can be decoded by hardware from macroblock header.

TYPE mb_coding_type of current macroblock; mb_coding_type can be decoded by hardware from mcbpc in macroblck header.

PATTERN pattern_code of current macroblock; pattern_code can be decoded by hardware from cbpc and cbpy in macroblock header.

DQUANT dquant. It can be –2, -1, +1 or +2; total 4 possible choices using 2 bits to represent; dquant can be decoded by hardware from macroblock header.

AC ac_pred_flag. It decides whether AC prediction is needed; always 0 in encoder; ac_pred_flag can be decoded by hardware from macroblock header.

DCVLC use_intra_dc_vlc. If this bit is 0, intra AC VLC decode is used (no intra DC exists in current macroblock).

QUANTIZER quantizer_scale, ranged from 1 to 31. It can be variable if we have dquant values.

MP4+0344h Core MB Structure 1 Register MP4_CORE_MB_STRUCT1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	DC[1]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	DC[0]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the DC value set 0 and 1 of current macroblock.

DC[0] DC Value for Luminance Block 0

DC[1] DC Value for Luminance Block 1

MP4+0348h Core MB Structure 2 Register MP4_CORE_MB_STRUCT2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	DC[3]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	DC[2]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the DC value set 2 and 3 of current macroblock. For debug purpose or SW encode/decode procedure.

DC[2] DC Value for Luminance Block 2

DC[3] DC Value for Luminance Block 3

MP4+034Ch Core MB Structure 3 Register MP4_CORE_MB_STRUCT3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	DC[5]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	DC[4]											
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the DC value set 4 and 5 of current macroblock. For debug purpose or SW encode/decode procedure.

DC[4] DC Value for Chrominance Block 4
DC[5] DC Value for Chrominance Block 5

MP4+0350h Core MB Structure 4 Register MP4_CORE_MB_STRUCT4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	MVY[0]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	MVX[0]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the motion vector set 0 of current macroblock. For debug purpose or SW encode/decode procedure.

MVX[0] X Component of Motion Vector Set 0
MVY[0] Y Component of Motion Vector Set 0

MP4+0354h Core MB Structure 5 Register MP4_CORE_MB_STRUCT5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	MVY[1]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	MVX[1]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the motion vector set 1 of current macroblock. For debug purpose or SW encode/decode procedure.

MVX[1] X Component of Motion Vector Set 1
MVY[1] Y Component of Motion Vector Set 1

MP4+0358h Core MB Structure 6 Register MP4_CORE_MB_STRUCT6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	MVY[2]							

Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	MVX[2]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the motion vector set 2 of current macroblock. For debug purpose or SW encode/decode procedure.

MVX[2] X Component of Motion Vector Set 2
MVY[2] Y Component of Motion Vector Set 2

MP4+035Ch Core MB Structure 7 Register MP4_CORE_MB_STRUCT7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	MVY[3]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	MVX[3]							
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to store the motion vector set 3 of current macroblock. For debug purpose or SW encode/decode procedure.

MVX[3] X Component of Motion Vector Set 3
MVY[3] Y Component of Motion Vector Set 3

5.23.2.6 VLC DMA

MP4+0370h Core VLC DMA Status Register MP4_CORE_VLC_DMA_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GADDR_LSB															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	GLCOMD	GDRDY	FULL	EMPTY	VLD	VLE	PACK	GREQ	STATE					
Type			RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register provides software program the information of current status of VLD DMA.

STATE State of VLC DMA Engine.
GREQ request for data read/write.
- 0 No request.
- 1 Request for data read/write.

PACK VLE Buffer Maximum Size Meet.
VLE VLE Stream Ready
- 0 VLE is not ready
- 1 VLE is ready

VLD VLD Stream Ready
- 0 VLD is not ready

1 VLD is ready

EMPTY FIFO Empty

0 VLC DMA FIFO is not empty

1 VLC DMA FIFO is empty

FULL FIFO Full

0 VLC DMA FIFO is not full

1 VLC DMA FIFO is full

GDRDY Waiting for gdrdy, a signal from GMC, to return.

0 gdrdy has been received.

1 Waiting for gdrdy to return.

GLCOMD Waitinf for glcomd, a signal from GMC, to return.

0 glcomd has been received.

1 Waiting for glcomd to return.

GADDR_LSB Lower 16 bit value of gaddr, a signal to GMC.

MP4+0374h Core VLE Status Register MP4_CORE_VLE_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	RELOAD_CNT									DONE
Type							RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

This register shows the status of MPEG4 VLE block and is used for hardware debugging.

DONE DC/AC coefficient reload done.

RELOAD_CNT DC/AC coefficient reload count.

MP4+0378h Core VLC DMA Base Address Register MP4_CORE_VLC_BASE_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASE															
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BASE														-	-
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO		

This register is used to describe the address of started Code Word for each VLC DMA buffer. Note that this base address should be 4-byte aligned.

BASE VLC DMA Base Address

MP4+037Ch Core VLC DMA Base Bit Count Register MP4_CORE_VLC_BASE_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	BIT				
Type												WO	WO	WO	WO	WO

This register is used to describe the starting bit position of the 1st Code Word in the 1st VLC DMA buffer. For the following VLC DMA buffers, it is assumed that they are all 4-byte aligned and always start from bit position "0".

BIT Start of Bit at the 1st Code Word of 1st DMA Buffer

MP4+0380h Core VLC DMA Buffer Limit Register MP4_CORE_VLC_LIMIT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LIMIT															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register is used to describe the buffer size of each VLC DMA buffer. Note that the value is counted in word (32-bit). Whenever the limit is reached and the corresponding interrupt control is enabled, an interrupt request will be generated.

LIMIT DMA Buffer Size, Count in Word (32-bit)

MP4+0384h Core VLC DMA Current Word Register MP4_CORE_VLC_WORD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADDR															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR														-	-
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO		

This register provides the address information of a certain code word that is under process of video CODEC. SW reads it back after encode of a frame is done.

ADDR VLC DMA current Address

MP4+0388h Core VLC DMA Current Bit Count Register MP4_CORE_VLC_BITCNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-	-	-	-	BITCNT				
Type												RO	RO	RO	RO	RO

This register provides the bit position information of a certain Code Word that is under process of video CODEC.

BITCNT Current Bit Count

MP4+038Ch Core VLC DMA Ring Buffer Ending Address Register MP4_CORE_VLC_JUMP_FROM_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_FROM_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_FROM_ADDR The ending address of the current DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR. To disable the ring buffer feature, set this register to all ones; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

MP4+0390h Core VLC DMA Ring Buffer Starting Address Register MP4_CORE_VLC_JUMP_TO_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	JUMP_TO_ADDR															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

JUMP_TO_ADDR The starting address of the next DMA buffer; when a jump takes place in VLC DMA address counter, the address will jump from the ending address of the current DMA buffer, which is JUMP_FROM_ADDR, to the starting address of the next DMA buffer, which is JUMP_TO_ADDR; note that the address counter will not jump until done with the content in memory with address as JUMP_FROM_ADDR. So the memory content with address JUMP_FROM_ADDR will be executed by hardware.

5.23.3 PNG Decoder

Overview

The PNG (Portable Network Graphics) images, is widely used in many applications like icons, picture, and so on. In order to display many true-color icons in our applications and boost image processing performance, the hardware PNG decoder is developed. As a result, PNG Decoder is designed to decode all kinds of PNG images with all color-depths combinations.

To gain the best speed performance, PNG decoder can handle the PNG file from the first start bits of IDAT-chunk till the end of this PNG file. The software program only needs to program related control registers

based on the IHDR chunk or sometimes PLTE chunk (palette mode) and wait for an interrupt coming from hardware. **Fig 1** shows the basic PNG file structure and software/hardware division. **Fig 2** shows the IHDR header that SW needs to parse.

The information of DQT and DHT table is included in the JPEG file but need to be parsed by the JPEG decoder and store in the memory. The software program must program 2K-byte-align address in the table starting address because we has fixed the locations of all kinds of tables, as shown in **Fig 2**.

Fig 3 shows the HW decoding procedures. From the first bit of IDCT Chunk, HW will first do the extracting of chunk data. HW then get the zlib compressed bitstream by attaching every IDAT DATA. After extracting LZ77 data from the zlib bitstream, HW will do read block-by-block in this LZ77 bitstream. In doing block decoding, HW support all types of LZ77 block decoding including uncompressed, dynamic Huffman, and fixed Huffman coded block. The coding type and final block information can be found in the first 3 bits of the block header. To decode the dynamic Huffman coded block, HW will first decode the HCLEN table, storing it into Uncompressed Memory (External Memory), then a compression action is performed to compress the table in external memory (128K) into Compressed memory (Internal Memory recommended) (60bytes). Same as the HCLEN table, the compression is performed for the HLIT and HDIST table. Because the compressed HDIST table born only after the HCLEN table can be discarded, SW can only allocate single HDIST compressed memory for both HCLEN and HDIST table storage. The compressed HLIT table needs 572 bytes, and the compressed HDIST table needs 60 bytes. Notes that if the PNG image has no dynamic Huffman coded block, these 3 memory (UNCOMPRESSED, HLIT, HCLEN) are no need to be allocated.

Fig. 4 shows the block diagram of the PNG HW decoder. Besides Uncompressed, HLIT, and HCLEN memory, the source file, LZ77 buffer, upper filtered buffer, color table, transparency table, and output memories are needed to be allocated and assigned by SW.

Fig. 5 shows the overall view of the PNG decoder and PNG resizer. PNG decoder sends the decoded image pixel-by-pixel to the PNG resizer in decoding a PNG image. The PNG resizer then doing the resizing, clipping, alpha blending and so on, then writes the resized image to the output buffer.

PNG decoder supports the pause/resume function. The pause/resume steps are described below:

(1) setting first infile_start_addr & infile_start_bitcnt for the first bits of IDAT chunk (the first infile_start_addr don’t need to be 4-bytes aligned)
(2) Set infile_end_addr for the pause point. This infile_end_addr should be infile_end_addr = (true_segment_boundary-8) | 0x3; where true_segment_boundary is the boundary which contains the true part of the PNG bitstream.
(3) Set PNG decode start, then PNG decoder starts decoding PNG file
(4) HW then sends an interrupt (flag2) when bitstream DMA reaches infile_end_addr and paused
(5) Set new infile_start_addr for the resume. infile_start_addr_new = (infile_start_addr_old & 0xfffffffc) + ((infile_consumed_bits>>3) & (0xfffffffc)); notes that the new inflie_start_addr would be 4-bytes aligned address.
(6) Set infile_consumed_bits to 0
(7) Clear IRQ status (flag2)
(8) Set PNG decode start, then PNG decoder resumes to decode.

PNG decoder sends an error interrupt when Sequencer parsing error (flag2), CRC checker fail (flag3), Error filter type (flag4), output file over out_file_end_addr (flag5), or Adler checker fail (flag6) occurred.

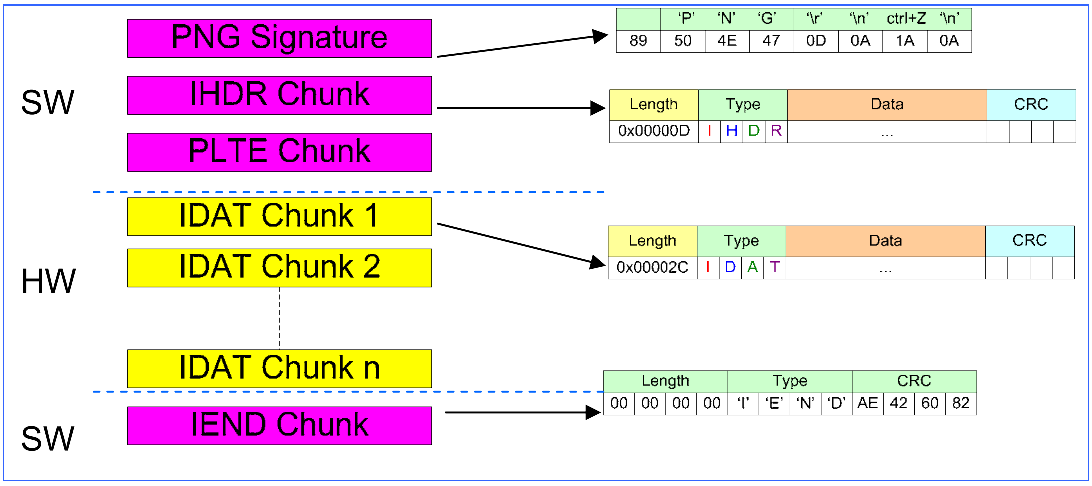


Fig. 18 The basic structure of PNG files.

Field Name	Size	Description
Width	4 bytes	Image width in pixels
Height	4 bytes	Image height in pixels
Bit Depth	1 byte	1,2,4,8 or 16
Color Type	1 byte	0:Grayscale, 2:RGB Triple, 3:Palette, 4:Grayscale+Alpha, 6: RGB+Alpha
Compression Method	1 byte	Must be 0
Filter Method	1 byte	Must be 0
Interlace Method	1 byte	0: non-interlaced. 1: Adam 7 interlacing

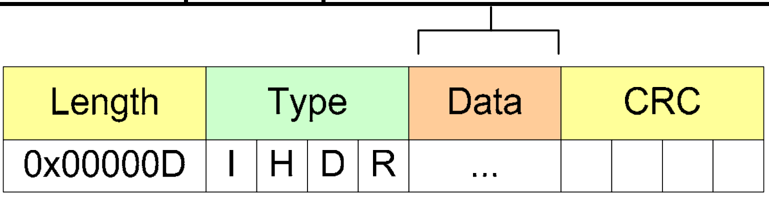


Fig. 19 IHDR header which SW needs to parse

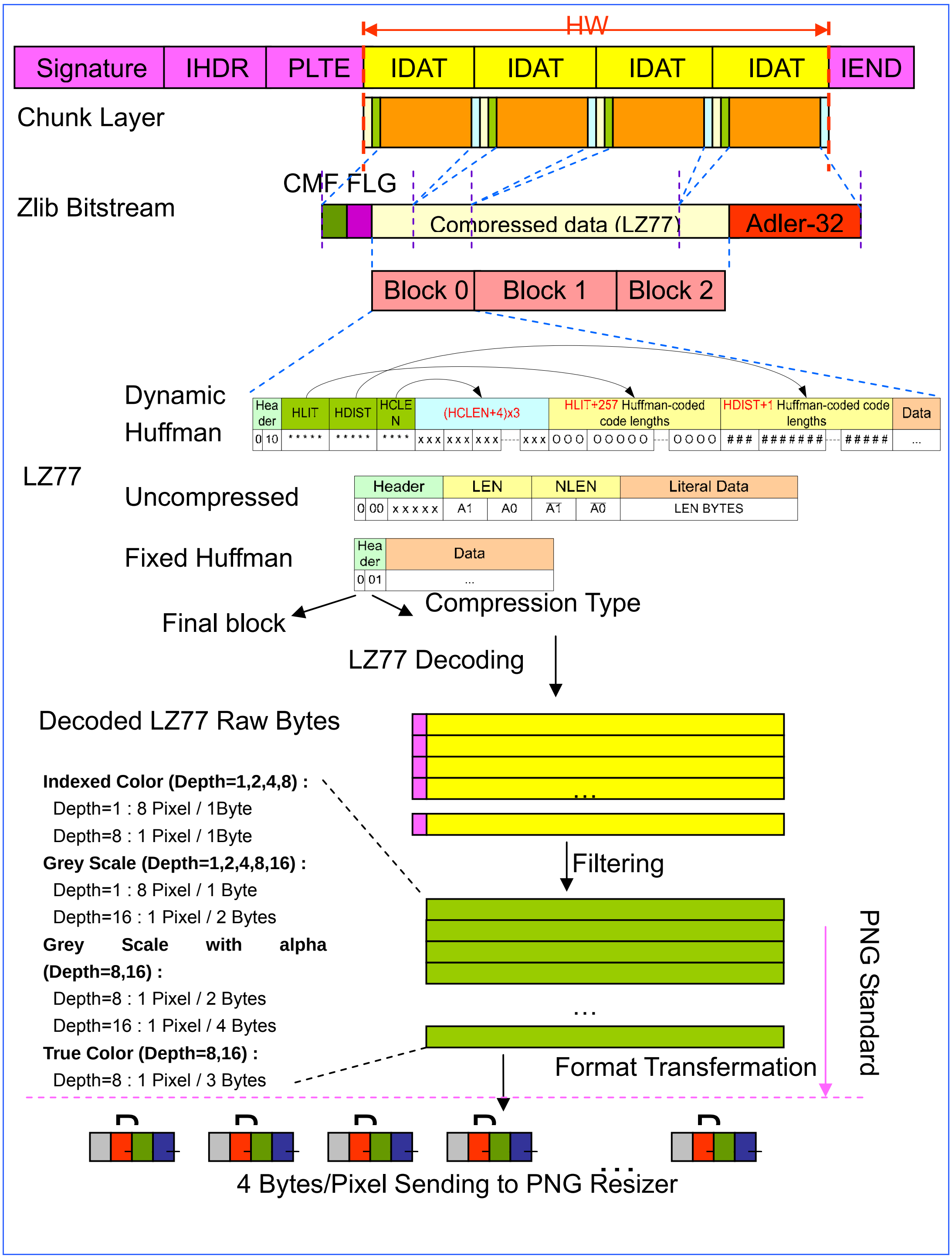


Fig. 20 The hardware decoding flow.

Fig. 21 Hardware block diagram.

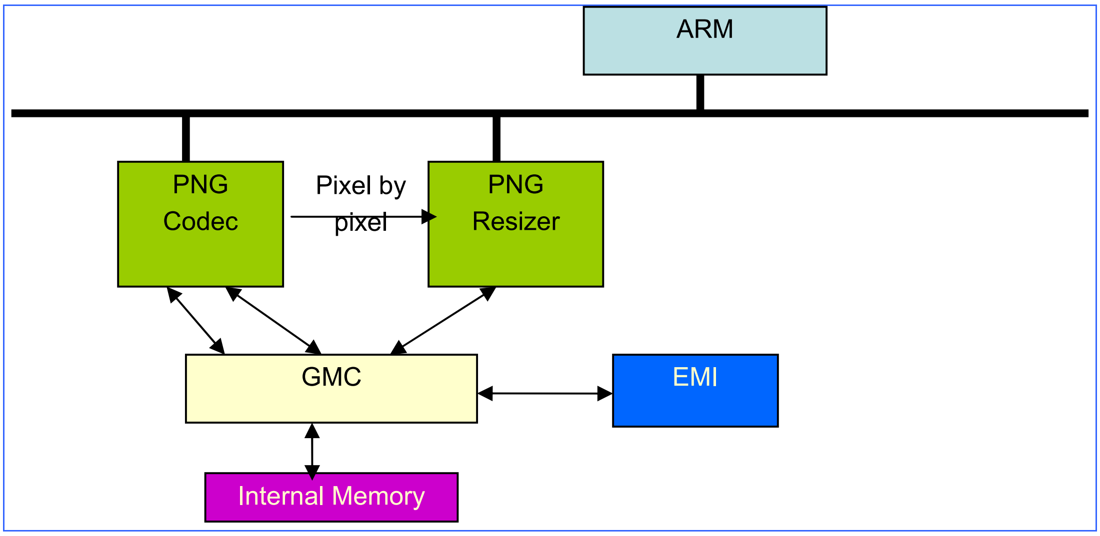


Fig. 22 The PNG decoder and PNG resizer.

5.23.4 Destination Working Buffer

The hardware will perform sizing and clipping function during storing output data to working buffer. The sizing algorithm is simply add-drop pixel. The HW will support arbitrary ratio of sizing. As shown below the output destination address is indicated by (dst_x,dst_y) and with a given size by dst_w and dst_h, in the mean term the source image is also given a size with src_w and src_h and the source image will be applied to sizing algorithm to enlarge or shorten the size to fit the destination image. We should note that the destination starting point may be negative. The output pixel in negative position won't be displayed on LCD window so neither output to working buffer. The clipping window is used to specify which part of image the LCD wants to display. The pixel out of clipping window should not output to working buffer. The destination working buffer is always start from (0.0) and with a given width (LCD_W) and height (LCD_H), if the pixel of destination image is in the clip window and LCD window it should be output to working buffer, and the address should be calculated as

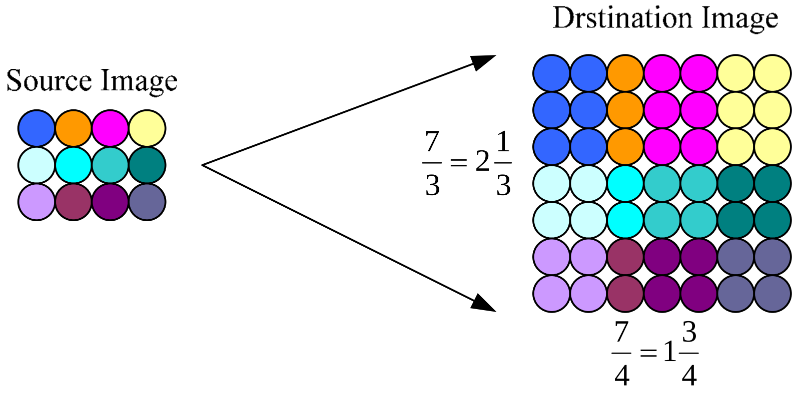
dst_output_addr = output_file_start_address + LCD_W * dst_y_ing * bytes_per_pixel + dst_x_ing * bytes_per_pixel

bytes_per_pixel will depend on output format and list as follow

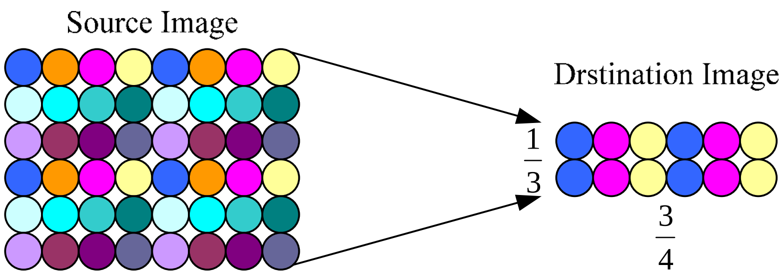
ARGB888 : 4 bytes/pixel
RGB888 : 3 bytes/pixel
RGB565 : 2 bytes/pixel
Unpacked index : 1 bytes/pixel

5.23.5 Sizing

1. Enlarge



2. Narrow



For simply hardware design HW need SW to provide these five parameters

$$1. ratio_width = \frac{dst_w}{src_w} = \mathrm{ratio_w_q} \cdot \frac{\mathrm{ratio_w_n}}{src_w}$$

$$2. ratio_height = \frac{dst_h}{src_h} = \mathrm{ratio_h_q} \cdot \frac{\mathrm{ratio_h_n}}{src_h}$$

$$3. ratio_height * 2 = \frac{dst_h * 2}{src_h} = \mathrm{ratio_h_q_mul_2} \cdot \frac{\mathrm{ratio_h_n_mul_2}}{src_h}$$

$$4. ratio_height * 4 = \frac{dst_h * 4}{src_h} = \mathrm{ratio_h_q_mul_4} \cdot \frac{\mathrm{ratio_h_n_mul_4}}{src_h}$$

$$5. ratio_height * 8 = \frac{dst_h * 8}{src_h} = \mathrm{ratio_h_q_mul_8} \cdot \frac{\mathrm{ratio_h_n_mul_8}}{src_h}$$

5.23.6 Limitation

The maximum decodable image size is

max_width=min(floor((16*1024-1)/bytes_per_pixel) , 8192);

max_height = is_interlace ? 8192 : 4096;

 © 2009 MediaTek Inc.

max_height * (max_width * bytes_per_pixel + 1) < 2^{23}
where bytes_per_pixel=8 for ARGB-16, 3 for RGB-8, 0.25 for Grey-4, 1 for palette, and etc.

5.23.7 Register Definitions

Register Address	Register Function	Acronym
PNGDEC+0000h	Input File Start Address	**INFILE_START_ADDR**
PNGDEC+0004h	Input File count	**INFILE_COUNT**
PNGDEC+0008h	Input File Consumed Bits	**INFILE_CONSUMED_BIT**
PNGDEC+000Ch	Color Table Start Address	**CT_START_ADDR**
PNGDEC+0010h	Uncompressed Start Address	**UNCOMPRESSED_START_ADDR**
PNGDEC+0014h	Huffman HLIT Table Start Address	**HLIT_START_ADDR**
PNGDEC+0018h	Huffman HDIST Table Start Address	**HDIST_START_ADDR**
PNGDEC+001Ch	Line Buffer0 Start Address	**BUFF0_START_ADDR**
PNGDEC+0020h	LZ77 Buffer Start Address	**LZ77_START_ADDR**
PNGDEC+0024h	Color_Type	**COLOR_TYPE**
PNGDEC+0028h	Decode Control Register	**DECODE_CTRL**
PNGDEC+002Ch	Clear Enable Register	**CLR_EN**
PNGDEC+0030h	ADLER	**ADLER**
PNGDEC+0034h	Color Output Format	**OUT_FORMAT**
PNGDEC+0038h	Interrupt Enable Register	**IRQ_EN**
PNGDEC+003Ch	Interrupt Status	**IRQ_STATUS**
PNGDEC+0040h	Transparency table start address	**TRNS_ADDR**
PNGDEC+0044h	TRNS CTRL	**TRNS CTRL**
PNGDEC+0048h	Transparency key1	**TRNS_key1**
PNGDEC+004Ch	Transparency key2	**TRNS_key2**
PNGDEC+0050h	Background Color	**BG_COLOR**
PNGDEC+0054h	INDEX NUMBER	**INDEX_NUM**
PNGDEC+0058h	Outfile start address	**OUTFILE_START_ADDR**
PNGDEC+05Ch	Outfile end address	**OUTFILE_END_ADDR**
PNGDEC+0060h	Source image size	**SRC_IMAGE_SIZE**
PNGDEC+0064h	Source image pixel count	**SRC_IMAGE_PIXEL_CNT**
PNGDEC+0068h	Destination image position	**DST_IMAGE_POSITION**
PNGDEC+006Ch	Destination sizing ration W1	**DST_SIZing_Ratio_w1**
PNGDEC+0070h	Destination sizing ration W2	**DST_SIZing_Ratio_w2**
PNGDEC+0074h	Destination sizing ration W4	**DST_SIZing_Ratio_w4**
PNGDEC+0078h	Destination sizing ration W8	**DST_SIZing_Ratio_w8**
PNGDEC+007Ch	Destination sizing ration H1	**DST_SIZing_Ratio_H1**

© 2009 MediaTek Inc.

PNGDEC+0080h	Destination sizing ration H2	**DST_SIZing_Ratio_h2**
PNGDEC+0084h	Destination sizing ration H4	**DST_SIZing_Ratio_h4**
PNGDEC+0088h	Destination sizing ration H8	**DST_SIZing_Ratio_h8**
PNGDEC+008Ch	LCD window size	**LCD_WINDOW_SIZE**
PNGDEC+0090h	Clip window position upper-left	**CLIP_WINDOW_POSITION_UL**
PNGDEC+0094h	Clip window position down-right	**CLIP_WINDOW_POSITION_DR**
PNGDEC+0098h	Interlace control	**INTERLACE_CTRL**
PNGDEC+009Ch	Replaced Color	**REPLACED_COLOR**
PNGDEC+00A0h	Replace As color	**REPLACE_AS_COLOR**
PNGDEC+00A4h	Alpha Blending	**ALPHA_BLENDING**
PNGDEC+00A8h	Alpha value	**A_VALUE**
PNGDEC+00ACh	B Buffer Start Address	**B_START_ADDR**
PNGDEC+00B0h	B buffer format	**B_FORMAT**
PNGDEC+0100h	PNG Debug information 1	**PNG_DEBUG1**
PNGDEC+0104h	PNG Debug information 2	**PNG_DEBUG2**
PNGDEC+0108h	PNG Debug information 3	**PNG_DEBUG3**
PNGDEC+010Ch	PNG Debug information 4	**PNG_DEBUG4**
PNGDEC+0110h	PNG Debug information 5	**PNG_DEBUG5**
PNGDEC+0114h	PNG Debug information 6	**PNG_DEBUG6**
PNGDEC+0118h	PNG Debug information 7	**PNG_DEBUG7**
PNGDEC+011Ch	PNG Debug information 8	**PNG_DEBUG8**
PNGDEC+0120h	PNG Debug information 9	**PNG_DEBUG9**
PNGDEC+0124h	PNG Debug information 10	**PNG_DEBUG10**

Table 119 ***PNG Decoder Registers***

PNGDEC+0000h Input File Start Address INFILE_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INFILE_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INFILE_START_ADDR[15:0]															
Type	R/W															
Reset	0															

INFILE_START_ADDR The input file starting address; PNG decoder would get decompression data from this address. The address can be any byte alignment.

PNGDEC+0004h Input File count INFILE_COUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INFILE_END_ADDR[31:16]															

Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INFILE_ END_ADDR [15:0]															
Type	R/W															
Reset	0															

INFILE_END_ADDR The end address of input file in bytes. If PNG decoder fetch data reach to this address, it will stop and turn off DEC_EN and notify SW by interrupt. If pause-resume mechanism is wanted, SW just needs to turn on the DEC_EN when re-programmed the INFILE_START_ADDR and INFILE_END_ADDR. GIF decoder will continue to decode using last states. If SW want to start a new decode process, it should set the CLR_EN = 1 then re-set it to 0 to clear PNG decoder's states. The CLR_EN is used to reset GIF decoder's internal states.

PNGDEC+0008h Input File Consumed Bits INFILE_CONSUMED_BIT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									INFILE_COUSUMED_BIT[23:16]							
Type									R/W							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INFILE_COUSUMED_BIT[15:0]															
Type	R/W															
Reset	0															

INFILE_CONSUMED_BIT The consumed bits by PNG hardware decoder and units in bit. This number is used to inform SW how many bits HW has been read from input file. SW should notice that when HW issue input file empty interrupt the INFILE_CONSUMED_BIT doesn't need to equal to INFILE_COUNT. In fact when input file empty interrupt has been raised it imply that the HW has been DMA in the INFILE_COUNT data but may not consume all of them so if SW don't clear the internal states the next time when DEC_EN is turn on HW will start to decode from the residual bits last time left.

PNGDEC+000Ch Color Table Start Address CT_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CT_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CT_START_ADDR [15:0]															
Type	R/W															
Reset	0															

CT_START_ADDR The color table starting address.
It needs to be word aligned. And each palette entry is one word.
Each word with following format: word[31:0]={8'b0,R,G,B}.
The maximum size needed is 256x4 bytes

PNGDEC+0010h Uncompressed Start Address UNCOMPRESSED_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	UNCOMPRESSED_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	UNCOMPRESSED _START_ADDR[15:0]															
Type	R/W															
Reset	0															

UNCOMPRESSED _START_ADDR Uncompressed start address

It needs to be word aligned.

This table has Maximum size 128k Bytes

PNGDEC+0014h Huffman HLIT Table Start Address HLIT_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HLIT_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HLIT_START_ADDR[15:0]															
Type	R/W															
Reset	0															

HLIT_START_ADDE Huffman literal code table starting address.

It needs to be word aligned.

This table has Maximum size 572 Bytes

PNGDEC+0018h Huffman HDIST Table Start Address HDIST_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HDIST_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDIST_START_ADDR[15:0]															
Type	R/W															
Reset	0															

HDIST_START_ADDR Huffman distance code table starting address.

It needs to be word aligned.

This table has Maximum size 60 bytes

PNGDEC+001Ch Line Buffer0 Start Address BUFF0_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUFF0_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUFF0_START_ADDR[15:0]															
Type	R/W															

Reset	0

BUFF0_START_ADDR Line buffer0 starting address (for de-filtering).

It needs to be word aligned.

This table has Maximum size ((source_width * bytes_per_pixel + 4) & 0xfffffffc)

where bytes_per_pixel=8 for ARGB-16, 3 for RGB-8, 0.25 for Grey-4, 1 for palette, and etc.

PNGDEC+0020h LZ77 Buffer Start Address LZ77_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LZ77_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LZ77_START_ADDR[15:0]															
Type	R/W															
Reset	0															

LZ77_START_ADDR LZ77 buffer starting address.

It needs to be word aligned.

The Maximum size = Max(32K+16 bytes, image size)

PNGDEC+0024h Color type COLOR_TYPE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									COLOR_DEPTH[4:0]					COLOR_TYPE [2:0]		
Type									R/W					R/W		
Reset									0					0		

COLOR_TYPE Indicate color type of PNG image.

0: grayscale

1: reserved

2: true color

3: palette

4: grayscale with alpha

5: reserved

6: true color with alpha

7: reserved

COLOR_DEPTH Indicate color bit depth of PNG image. 1, 2, 4, 8, 16

PNGDEC+0028h Decode Control Register DECODE_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

 © 2009 MediaTek Inc.

Name																DEC_EN
Type																R/W
Reset																0

DEC_EN Decode enable signal.

PNGDEC+002Ch Clear Enable Register clr_en

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CLR_EN
Type																R/W
Reset																0

CLR_EN Clear enable signal. 1 = clear, 0 = no clear

Note: To reset the PNG decoder, the CLR_EN should be first set to 1 then set to 0.

PNGDEC+0030h ADLER ADLER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ADLER[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADLER[15:0]															
Type	R/W															
Reset	0															

ADLER Adler checksum data report.

PNGDEC+0034h Color Output Format OUT_FORMAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															OUT_FORMAT[1:0]	
Type															R/W	
Reset															0	

OUT_FORMAT Output color format

2'b00: Index (Not Support)

2'b01: RGB565,

2'b10: RGB888,

2'b11: ARGB8888

PNGDEC+0038h Interrupt Enable Register IRQ_EN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

 © 2009 MediaTek Inc.

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										INT6_EN	INT5_EN	INT4_EN	INT3_EN	INT2_EN	INT1_EN	INT0_EN
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

INT0_EN Flag0 interrupt enable
INT1_EN Flag1 interrupt enable
INT2_EN Flag2 interrupt enable
INT3_EN Flag3 interrupt enable
INT4_EN Flag4 interrupt enable
INT5_EN Flag5 interrupt enable
INT6_EN Flag6 interrupt enable

PNGDEC+003Ch Interrupt Status IRQ_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset										0	0	0	0	0	0	0

FLAG0 Image decompressed complete decoded pixels exactly the same as image size
FLAG1 Image decompressed complete result from Input file empty
FLAG2 Image decompressed incomplete result from sequencer error
FLAG3 Image decompressed incomplete result from CRC error
FLAG4 Image decompressed incomplete result from decoder decode an error filter type
FLAG5 Image decompressed incomplete result from output address over out_end_addr
FLAG6 Image decompressed incomplete result from Adler Error

PNGDEC+0040h Transparency table start address TRNS_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TRNS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TRNS_ADDR[15:0]															
Type	R/W															
Reset	0															

TRAN_ADDR transparency table start address.

PNGDEC+0044h TRNS CTRL TRNS CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													T_OUT_SPEC	T_TABLE	T_OUT	T_EN
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

T_EN Transparent enable
T_OUT 0: output transparent color as background color 1: no output when transparent
T_TABLE Transparent table exist
T_OUT_SPEC Transparent color key enable

PNGDEC+0048h Transparency key1 TRNS_key1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GREY_KEY[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	R_KEY[15:0]															
Type	R/W															
Reset	0															

GREY_KEY Transparent color key of grayscale image
R_KEY Transparent color key of red component

PNGDEC+004Ch Transparency key2 TRNS_key2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G_KEY[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_KEY[15:0]															
Type	R/W															
Reset	0															

G_KEY Transparent color key of green component
B_KEY Transparent color key of blue component

PNGDEC+0050h Background Color BG_COLOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BG_GREY[7:0]								BG_R[7:0]							
Type	R/W								R/W							
Reset	0								0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BG_G[7:0]								BG_B[7:0]							
Type	R/W								R/W							
Reset	0								0							

BG_GREY background color of grayscale image
BG_R background color of red component
BG_G background color of green component

BG_B background color of blue component

PNGDEC+0054h INDEX NUMBER INDEX_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								COLOR_NUM[7:0]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TRANS_NUM[7:0]								
Type								R/W								
Reset								0								

COLOR_NUM color entry number
TRANS_NUM transparency entry number

PNGDEC+0058h Outfile start address OUTFILE_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OUTFILE_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OUTFILE_START_ADDR[15:0]															
Type	R/W															
Reset	0															

OUTFILE_START_ADDR Output working buffer address we can view this buffer as LCD display buffer. This address is viewed as the (0,0) address of LCD window. It needs word aligned.

PNGDEC+05Ch Outfile end address OUTFILE_END_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OUTFILE_END_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OUTFILE_END_ADDR[15:0]															
Type	R/W															
Reset	0															

OUTFILE_END_ADDR Output file end address. The end address is set to prevent gif decoder from writing the wrong memory sections. When this happens, PNG decoder will stop and generate an interrupt to inform SW. It needs word aligned.

PNGDEC+0060h Source image size SRC_IMAGE_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRC_W[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC_H[15:0]															
Type	R/W															
Reset	0															

 © 2009 MediaTek Inc.

SRC_W The width of the source image specifies in pixel. This value is an un-signed 16 bits. This parameter is used when sizing function required. The usage is described in the above section. The range is from 1~2^16-1. 0 is not valid.

SRC_H The height of the source image specifies in pixel. This value is an un-signed 16 bits. This parameter is used when sizing function required. The usage is described in the above section. The range is from 1~2^16-1. 0 is not valid.

PNGDEC+0064h Source image pixel count SRC_IMAGE_PIXEL_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRC_IMAGE_PIXEL_CNT[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRC_IMAGE_PIXEL_CNT[15:0]															
Type	R/W															
Reset	0															

SRC_IMAGE_PIXEL_CNT the result of SRC_W * SRC_H, unit in pixel, using this register to save a multiplier.

PNGDEC+0068h Destination image position DST_IMAGE_POSITION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DST_X[15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_Y[15:0]															
Type	R/W															
Reset	0															

DST_X The x-position of the destination image specifies in pixel. This value is a signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. This value must be a 2's complement number. Its range is from -2^15 ~ 2^15-1

DST_Y The y-position of the destination image specifies in pixel. This value is a signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. This value must be a 2's complement number. Its range is from -2^15 ~ 2^15-1

PNGDEC+006Ch Destination sizing ration W1 DST_SIZING_RATIO_W1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									DST_W_Q[7:0]							
Type									R/W							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_W_N[15:0]															
Type	R/W															
Reset	0															

DST_W_Q The quotient of dst_w/src_w. This value is an un-signed 8 bits.
If DST_W_Q = 0, it means the resizer should perform shrink otherwise it would be enlarge function.

DST_W_N The remainder of dst_w/src_w.

PNGDEC+0070h Destination sizing ration W2 DST_SIZING_RATIO_W2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								DST_W_Q_MUL_2[8:0]								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_W_N_MUL_2[15:0]															
Type	R/W															
Reset	0															

DST_W_Q_MUL_2 The quotient of 2*dst_w/src_w. This value is an un-signed 9 bits.

DST_W_N_MUL_2 The remainder of 2*dst_w/src_w. This value is an un-signed 16 bits.

PNGDEC+0074h Destination sizing ration W4 DST_SIZING_RATIO_W4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							DST_W_Q_MUL_4[9:0]									
Type							R/W									
Reset							0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_W_N_MUL_4[15:0]															
Type	R/W															
Reset	0															

DST_W_Q_MUL_4 The quotient of 4*dst_w/src_w. This value is an un-signed 10 bits.

DST_W_N_MUL_4 The remainder of 4*dst_w/src_w. This value is an un-signed 16 bits.

PNGDEC+0078h Destination sizing ration W8 DST_SIZING_RATIO_W8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						DST_W_Q_MUL_8[10:0]										
Type						R/W										
Reset						0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_W_N_MUL_8[15:0]															
Type	R/W															
Reset	0															

DST_W_Q_MUL_8 The quotient of 8*dst_w/src_w. This value is an un-signed 11 bits.

DST_W_N_MUL_8 The remainder of 8*dst_w/src_w. This value is an un-signed 16 bits.

PNGDEC+007Ch Destination sizing ration H1 DST_SIZING_RATIO_H1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									DST_H_Q[7:0]							
Type									R/W							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_H_N[15:0]															
Type	R/W															
Reset	0															

DST_H_Q The quotient of dst_h/src_h. This value is an un-signed 8 bits.

If DST_W_Q = 0, it means the resizer should perform shrink otherwise it would be enlarge function.

DST_H_N The remainder of dst_h/src_h.

PNGDEC+0080h Destination sizing ration H2 DST_SIZING_RATIO_H2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								DST_H_Q_MUL_2[8:0]								

Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_H_N_MUL_2[15:0]															
Type	R/W															
Reset	0															

DST_H_Q_MUL_2 The quotient of 2*dst_h/src_h. This value is an un-signed 9 bits.

DST_H_N_MUL_2 The remainder of 2*dst_h/src_h. This value is an un-signed 16 bits.

PNGDEC+0084h Destination sizing ration H4 DST_SIZING_RATIO_H4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							DST_H_Q_MUL_4[9:0]									
Type							R/W									
Reset							0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_H_N_MUL_4[15:0]															
Type	R/W															
Reset	0															

DST_H_Q_MUL_4 The quotient of 4*dst_h/src_h. This value is an un-signed 10 bits.

DST_H_N_MUL_4 The remainder of 4*dst_h/src_h. This value is an un-signed 16 bits.

PNGDEC+0088h Destination sizing ration H8 DST_SIZING_RATIO_H8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						DST_H_Q_MUL_8[10:0]										
Type						R/W										
Reset						0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_H_N_MUL_8[15:0]															
Type	R/W															
Reset	0															

DST_H_Q_MUL_8 The quotient of 8*dst_h/src_h. This value is an un-signed 11 bits.

DST_H_N_MUL_8 The remainder of 8*dst_h/src_h. This value is an un-signed 16 bits.

PNGDEC+008Ch LCD window size LCD_WINDOW_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LCD_W [15:0]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LCD_H [15:0]															
Type	R/W															
Reset	0															

LCD_W The width of the LCD window specifies in pixel. This value is an un-signed 16 bits. This parameter is used when sizing function required. The usage is described in the above section. The range is from 1~2^16-1. 0 is not valid.

LCD_H The height of the LCD window specifies in pixel. This value is an un-signed 16 bits. This parameter is used when sizing function required. The usage is described in the above section. The range is from 1~2^16-1. 0 is not valid.

PNGDEC+0090h Clip window position upper-left CLIP_WINDOW_POSITION_UL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CLIP_UL_X[15:0]**															
Type	**R/W**															
Reset	**0**															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CLIP_UL_Y[15:0]**															
Type	**R/W**															
Reset	**0**															

CLIP_UL_X The up-left x-position of the clip window specifies in pixel. This value is an un-signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. Its range is from 0 ~ 2^16-1

CLIP_UL_Y The up-left y-position of the clip window specifies in pixel. This value is an un-signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. Its range is from 0 ~ 2^16-1

PNGDEC+0094h Clip window position down-right CLIP_WINDOW_POSITION_DR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CLIP__DR_X[15:0]**															
Type	**R/W**															
Reset	**0**															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CLIP_DR_Y[15:0]**															
Type	**R/W**															
Reset	**0**															

CLIP_DR_X The down-right x-position of the clip window specifies in pixel. This value is an un-signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. Its range is from 0 ~ 2^16-1

CLIP_DR_Y The down-right y-position of the clip window specifies in pixel. This value is an un-signed 16 bits. This parameter is used to speed up gif display and reduce memory usage. Its range is from 0 ~ 2^16-1

PNGDEC+0098h Interlace control INTERLACE_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**INTERLACE**
Type																**R/W**
Reset																**0**

INTERLACE 0 - Non-interlaced 1 – Interlaced

PNGDEC+009Ch Replaced Color REPLACED_COLOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name								REPLACE_EN	REPLACED_COLOR[23:16]							
Type								R/W	R/W							
Reset								0	0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REPLACED_COLOR[15:0]															
Type	R/W															
Reset	0															

REPLACE_EN enable for replace function (when out_format is ARGB it should not be turn on)

REPLACED_COLOR The valid bit length is depend on out_format RGB888, replace color is 24 bits with RGB, RGB565, replace color is 16 bits

PNGDEC+00A0h Replace As color REPLACE_AS_COLOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									REPLACE_AS COLOR[23:16]							
Type									R/W							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REPLACE_AS COLOR[15:0]															
Type	R/W															
Reset	0															

REPLACE_AS_COLOR The format is depend on out_format, RGB888, replace color is 24 bits with RGB. RGB565, replace color is 16 bits

PNGDEC+00A4h Alpha Blending ALPHA_BLENDING

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													ALPHA_SIDE	BLENDING_MODE[2:0]		
Type													R/W	R/W		
Reset													0	0		

ALPHA_SIDE apply alpha in which side (a*r1_value+(1-a)*r2_value)

0: r1_value = PNG FILE, r2_value = B Buffer

1: r1_value = B buffer , r2_value = PNG FILE

BLENDING_MODE 3'b000: no blending (turn off blending)

3'b001: use PNG alpha value (only support when PNG format is RGB or grey with alpha) and blend with B buffer

3'b010: use B buffer alpha value (only support when B buffer is ARGB mode)

3'b011: use user defined src alpha value

3'b100: use PNG alpha value (only support when PNG format is RGB or grey with alpha) and blend with back ground color

Note:

Alpha Blending should be turned on only when PNG file is RGB with alpha or grey with alpha, if it turn on when PNG file hasn't alpha the transparent function may be failed.

PNGDEC+00A8h Alpha value A_VALUE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																A_EN
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DST_A_VALUE[7:0]								SRC_A_VALUE[7:0]							
Type	R/W								R/W							
Reset	0								0							

A_EN 0: using PNG FILE alpha value as output alpha value if output format is ARGB8888
(if our_format = ARGB888 only no blend act can turn off a_en)
1: using DST_A_VALUE as output alpha value if output format is ARGB8888

DST_A_VALUE Software specified value for using when output format is ARGB8888 and following two situations happened
Blending acts
PNG source FILE hasn't alpha value

SRC_A_VALUE This value is used for alpha blending when BLENDING_MODE == 2'b11.

PNGDEC+00ACh B Buffer Start Address B_START_ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	B_START_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	B_START_ADDR[15:0]															
Type	R/W															
Reset	0															

B_START_ADDR The b buffer start address and its coordination is (0,0)

PNGDEC+00B0h B buffer format B_FORMAT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															B_FORMAT[1:0]	
Type															R/W	
Reset															0	

B_FORMAT 2'b00: No deinfed
2'b01: RGB565
2'b10: RGB888
2'b11: ARGB8888

PNGDEC+0100h PNG Debug information 1 PNG_DEBUG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											SEQ_STATE[5:0]					
Type											R					
Reset											0					

SEQ_STATE sequencer state machine

PNGDEC+0104h PNG Debug information 2 PNG_DEBUG2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		VLD_CODEWORD[14:0]														
Type		R														
Reset		0														

VLD_CODEWORD vld_codeword

PNGDEC+0108h PNG Debug information 3 PNG_DEBUG3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				OUT_GMC_STATE[12:0]												
Type				R												
Reset				0												

OUT_GMC_STATE out state machine

PNGDEC+010Ch PNG Debug information 4 PNG_DEBUG4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name														PAL_WAIT_GDRDY	OUT_BUF_FULL	PIXEL_VALID
Type														R	R	R
Reset														0	0	0

PIXEL_VALID pixel_valid signal

OUT_BUF_FULL out_buf_full signal

PAL_WAIT_GDRDY pal gmc port wait gdrdy signal

PNGDEC+0110h PNG Debug information 5 PNG_DEBUG5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		RAW_DATA_X[14:0]														
Type		R														
Reset		0														
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					RAW_DATA_Y[11:0]											
Type					R											
Reset					0											

RAW_DATA_X LZ77 decoding position, x

RAW_DATA_Y LZ77 decoding position, y

PNGDEC+0114h PNG Debug information 6 PNG_DEBUG6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												IRQ_B		FILTER_TYPE		

Type												R		R		
Reset												0		0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ZLIB_BLOCKNO[1:0]**															
Type	**R**															
Reset	**0**															

ZLIB_BLOCKNO current zlib_block number

FILTER_TYPE filter_type

IRQ_B irq

PNGDEC+0118h PNG Debug information 7 PNG_DEBUG7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**DMA_STATE[4:0]**				
Type												**R**				
Reset												**0**				

DMA_STATE dma_state_machine

PNGDEC+011Ch PNG Debug information 8 PNG_DEBUG8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CRC[31:16]**															
Type	**R**															
Reset	**0**															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CRC[15:0]**															
Type	**R**															
Reset	**0**															

CRC current CRC value

PNGDEC+0120h PNG Debug information 9 PNG_DEBUG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**CRC_FIFO_BYTEPOS[4:0]**				
Type												**R**				
Reset												**0**				

CRC_FIFO_BYTEPOS crc_fifo byte position

PNGDEC+0124h PNG Debug information 10 PNG_DEBUG10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CRC_READ_LENGTH[31:16]**															
Type	**R**															
Reset	**0**															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CRC_READ_LENGTH[15:0]**															
Type	**R**															
Reset	**0**															

CRC_READ_LENGTH crc read length

5.24 Post Resize

5.24.1 General Description

Figure 116 shows the block diagram of post resize. It receives image data from a block-based source such as JPEG decoder or from a scan line based source, and then performs image resizing. The capability of resizing in the block is divided into two portions, coarse pass and fine pass. The first pass is coarse resizing pass and it is able to shrink image by a factor of 1, 1/4, 1/16, or 1/64. The second pass is the fine resizing pass, which is composed of horizontal and vertical resizing, and it is able to shrink or enlarge image in fractional ratio. The maximum allowable image size for the fine resizing pass is 4095x4095. Thus the maximum allowable image size for coarse resizing pass is 32760x32760 if the strip buffer is enough.

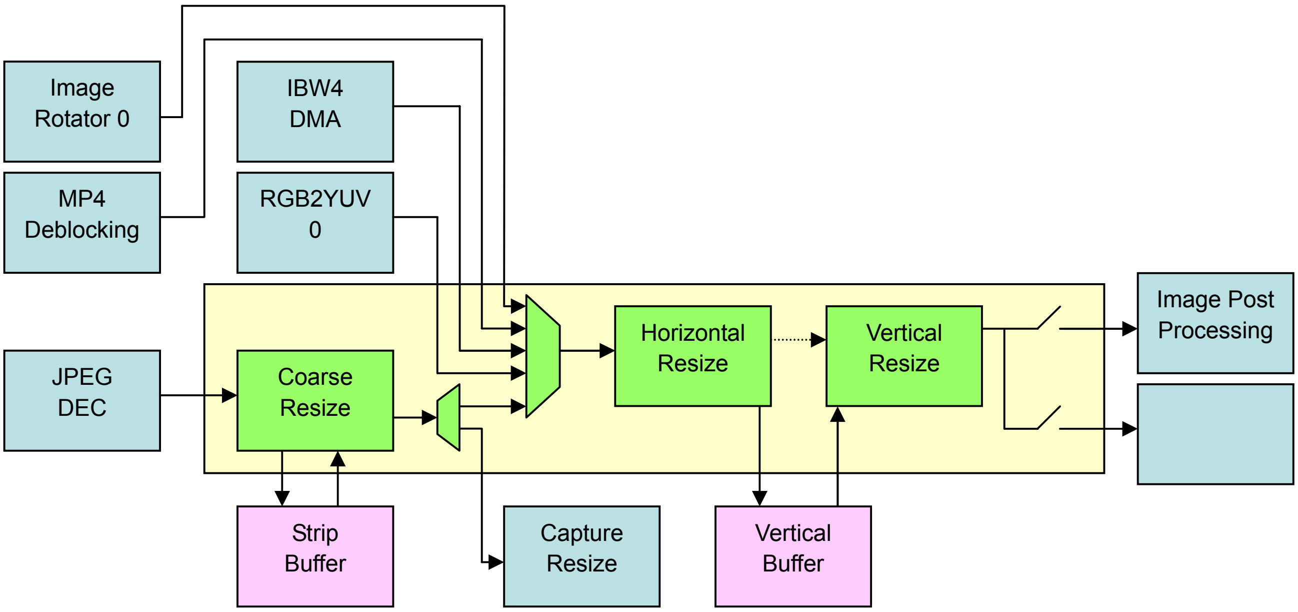


Figure 144 ***Block diagram of the post resize***

The strip buffer of coarse resizing pass accumulates de-compressed 8x8 YUV blocks separately. These YUV data are packed into pixels and sent to the fine resizing pass.

The fine resizing pass is composed of horizontal and vertical resizing blocks. It can scale up or down the input image by any ratio. However, the maximum sizes of input and output images are limited to 4095x4095. The horizontal resizing function is a combination of 2's power average and bi-linear interpolation. The vertical resizing function is a bi-linear interpolation. The input and output format are both YUV444. But the internal working memory format is YUV422 to mitigate memory and bandwidth requirements.

5.24.2 Working Memories

There are two working memories in post resize. One is the strip buffer, and the other is the vertical buffer.

5.24.2.1 Strip Buffer

Let's denote sampling factor for Y-component as (H_Y, V_Y), U-component as (H_U, V_U) and V-component as (H_V, V_V) in a JPEG file. The minimum requirement of memory size for the strip buffer is **(the image width of after the coarse resizing pass) * (V_Y * 8 + V_U * 8 + V_V * 8)** bytes. It is (4064) * (4 * 8 + 2 * 8 + 2 * 8) = 260K bytes

 © 2009 MediaTek Inc.

for extreme cases. To enhance the throughput of JPEG decode process, software may use double buffer scheme. Then it becomes 520K bytes. Please note that the strip buffer is composed of the Y buffer, U buffer, and V buffer. Software can allocate separate memory for them.

5.24.2.2 Vertical Buffer

The minimum requirement of memory for the vertical buffer is **(the target image width) * (the line number for vertical interpolation) * 2** bytes. It is (4095) * (2) * 2 = 16K bytes for extreme cases. To enhance throughputs of overall data paths, software may use double buffer scheme. Then it becomes 32K bytes.

5.24.3 Source Image

For the coarse resizing pass, the width of the source image must be multiples of **8 * (maximum horizontal sampling factor)**. Similarly, the height of the source image must be multiples of **8 * (maximum vertical sampling factor)**. The maximum size of target image is 4095x4095.
For the fine resizing pass, the maximum size of source image and target image are both 4095x4095.

5.24.4 Flow Control

For the coarse resizing pass, the coarse resizing will send pixel data to the fine resizing when they are ready with hand shake signal. If strip buffer is full, the coarse resizing will halt image data input until the strip buffer is available.
For the fine resizing pass, the fine resizing will send pixel data to the image post processing or yuv2rgb1 when they are ready with hand shake signal. If vertical buffer is full, the fine resizing will halt image data input until the vertical is available.

5.24.5 Register Definitions

REGISTER ADDRESS	REGISTER NAME	SYNONYM
PRZ+ 0000h	Post Resize Configuration Register	PRZ_CFG
PRZ + 0004h	Post Resize Control Register	PRZ_CON
PRZ + 0008h	Post Resize Status Register	PRZ_STA
PRZ + 000Ch	Post Resize Interrupt Register	PRZ_INT
PRZ + 0010h	Post Resize Source Image Size Register 1	PRZ_SRCSZ1
PRZ + 0014h	Post Resize Target Image Size Register 1	PRZ_TARSZ1
PRZ + 0018h	Post Resize Horizontal Ratio Register 1	PRZ_HRATIO1
PRZ + 001Ch	Post Resize Vertical Ratio Register 1	PRZ_VRATIO1
PRZ + 0020h	Post Resize Horizontal Residual Register 1	PRZ_HRES1
PRZ + 0024h	Post Resize Vertical Residual Register 1	PRZ_VRES1
PRZ + 0030h	Post Resize Block Coarse Shrinking Configuration Register	PRZ_BLKCSCFG
PRZ + 0034h	Post Resize Y-Component Line Buffer Memory Base Address	PRZ_YLMBASE
PRZ + 0038h	Post Resize U-Component Line Buffer Memory Base Address	PRZ_ULMBASE
PRZ + 003Ch	Post Resize V-Component Line Buffer Memory Base Address	PRZ_VLMBASE
PRZ + 0040h	Post Resize Fine Resizing Configuration Register	PRZ_FRCFG
PRZ + 0050h	Post Resize Y Line Buffer Size Register	PRZ_YLBSIZE

© 2009 MediaTek Inc.

PRZ + 005Ch	Post Resize Pixel-Based Resizing Working Memory Base Address	PRZ_PRWMBASE

5.24.5.1 Post Resize Configuration Register

PRZ+0000h Post Resize Configuration Register PRZ_CFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									LBSEL		PSEL	PCON	PELSRC1			
Type									R/W		R/W	R/W	R/W			
Reset									0		0	0	0000			

The register is for global configuration of Post Resize.

PELSRC1 The register field specifies which pixel-based image source is serviced.

- **1** Image Rotator 0
- **2** MPEG4 deblocking
- **3** IBW4 DMA
- 5 R2Y 0
- 6 JPEG Decoder
- **Others** Reserved

PCON The register bit specifies if pixel-based resizing continues whenever an image finishes processing. If immediate stop is desired, reset Post Resize directly. If the last image is desired, set the register bit to ‘0’ first. Then wait till Post Resize is not busy again. Finally reset Post Resize.

- **0** Single run
- **1** Continuous run

PSEL The register field determines if block-based image sources are serviced.

- **0** PELSRC1 is JPEG Decoder.
- **1** PELSRC1 is not JPEG Decoder.

LBSEL line buffer selection. **When CRZ_CFG.LB_SEL is set to 1, this bit should not be set to 1.**

- **0** Use shared memory as temporary buffer.
- **1** Use RESZ_LB (resizer dedicated line buffer) as temporary buffer. Total 24KB.

5.24.5.2 Post Resize Control Register

PRZ+0004h Post Resize Control Register PRZ_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														PELVRRST	PELHRRST	BLKCSRST
Type														R/W	R/W	R/W
Reset														0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PELVRENA	PELHRENA	BLKCSENA
Type														R/W	R/W	R/W
Reset														0	0	0

The register is for global control of Post Resize. **Note that block-based and pixel-based resizing cannot execute parallel. Furthermore, software reset will NOT reset all register setting. Remember trigger Post Resize first before trigger image sources to Post Resize.**

BLKCSENA Writing '1' to the register bit will cause Block Coarse Shrinking proceed to work. Block Coarse Shrinking is designed to cooperate width JPEG decoder. It works on the fly. Bu it needs to be restarted every time before working.

PELHRENA Writing '1' to the register bit will cause pixel-based fine horizontal resizing proceed to work. However, if horizontal resizing is not necessary, do not write '1' to the register bit.

PELVRENA Writing '1' to the register bit will cause pixel-based fine vertical resizing proceed to work. However, if vertical resizing is not necessary, do not write '1' to the register bit.

BLKCSRST Writing '1' to the register bit will force Block Coarse Shrinking to stop immediately and have Block Coarse Shrinking keep in reset state. In order to have Block Coarse Shrinking go to normal state, writing '0' to the register bit.

PELHRRST Writing '1' to the register will cause pixel-based fine horizontal resizing to stop immediately and have pixel-based fine horizontal resizing keep in reset state. In order to have pixel-based fine horizontal resizing go to normal state, writing '0' to the register bit.

PELVRRST Writing '1' to the register will pixel-based fine vertical resizing to stop immediately and have pixel-based fine vertical resizing keep in reset state. In order to have pixel-based fine vertical resizing go to normal state, writing '0' to the register bit.

5.24.5.3 Post Resize Status Register

PRZ+0008h Post Resize Status Register PRZ_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name												BLKINTRABSY		PELVRBUSY	PELHRBUSY	BLKCSBUSY
Type												RO		RO	RO	RO
Reset												0		0	0	0

The register indicates global status of Post Resize.

BLKCSBUSY Block-based CS (Coarse Shrinking) Busy Status

PELHRBUSY Pixel-based HR (Horizontal Resizing) Busy Status

PELVRBUSY Pixel-based VR (Vertical Resizing) Busy Status

BLKINTRABSY Block-based CS (Coarse Shrinking) Intra-Block Busy Status

5.24.5.4 Post Resize Interrupt Register

PRZ+000Ch Post Resize Interrupt Register PRZ_INT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PELVRINT	PELHRINT	BLKCSINT
Type														RC	RC	RC
Reset														0	0	0

The register shows up the interrupt status of resizer.

BLKCSINT Interrupt for BLKCS (Block-based Coarse Shrink). No matter if the register bit PRZ_BLKCSCFG.INTEN is enabled or not, the register bit will be active whenever BLKCS completes. It could be used as software interrupt by polling the register bit. Clear it by reading the register.

PELHRINT Interrupt for PELHR (Pixel-based Horizontal Resizing). No matter if the register bit PRZ_FRCFG.HRINTEN is enabled or not, the register bit will be active whenever PELHR completes. It could be used as software interrupt by polling the register bit. Clear it by reading the register.

PELVRINT Interrupt for PELVR (Pixel -based Vertical Resizing). No matter if the register bit PRZ_FRCFG.VRINTEN is enabled or not, the register bit will be active whenever PELVR completes. It could be used as software interrupt by polling the register bit. Clear it by reading the register.

5.24.5.5 Post Resize Source Image Size Register 1

PRZ+0010h Post Resize Source Image Size Register 1 PRZ_SRCSZ1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					HS											
Type					R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					WS											
Type					R/W											

The register specifies the size of source image after coarse shrink process. **The maximum allowable size is 4095x4095**. Note that for the width of source image must be multiples of 8xH_{max} and the height of source image must be multiples of 8xV_{max} when Block Coarse Shrinking is involved. And even if PRZ is set to output to CRZ directly (PRZ_FRCFG.O_CRZ = 1), WS and HS should be filled as Block Coarse Shrinking result.

WS The register field specifies the width of source image after coarse shrink process.

- **1** The width of source image after coarse shrink process is 1.
- **2** The width of source image is 2.
- …

HS The register field specifies the height of source image after coarse shrink process.

- **1** The height of source image after coarse shrink process is 1.
- **2** The height of source image after coarse shrink process is 2.
- …

5.24.5.6 Post Resize Target Image Size Register 1

PRZ+0014h Post Resize Target Image Size Register 1 PRZ_TARSZ1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**HT**											
Type					R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**WT**											
Type					R/W											

The register specifies the size of target image. **The maximum allowable size is 4095x4095.**

WT The register field specifies the width of target image.

- **1** The width of target image is 1.
- **2** The width of target image is 2.
- …

HT The register field specifies the height of target image.

- **1** The height of target image is 1.
- **2** The height of target image is 2.
- …

5.24.5.7 Post Resize Horizontal Ratio Register 1

PRZ+0018h Post Resize Horizontal Ratio Register PRZ_HRATIO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RATIO [31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RATIO [15:0]**															
Type	R/W															

The register specifies horizontal resizing ratio. It is obtained by PRZ_SRCSZ.WS * 2^{20} / PRZ_TARSZ.WT.

5.24.5.8 Post Resize Vertical Ratio Register 1

PRZ+001Ch Post Resize Vertical Ratio Register 1 PRZ_VRATIO1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RATIO [31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RATIO [15:0]**															
Type	R/W															

The register specifies vertical resizing ratio. It is obtained by PRZ_SRCSZ.HS * 2^{20} / PRZ_TARSZ.HT.

5.24.5.9 Post Resize Horizontal Residual Register 1

PRZ+0020h Post Resize Horizontal Residual Register 1 PRZ_HRES1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESIDUAL**															
Type	R/W															

The register specifies horizontal residual. It is obtained by PRZ_SRCSZ.WS % PRZ_TARSZ.WT. The maximum allowable value is 4094.

5.24.5.10 Post Resize Vertical Residual Register 1

PRZ+0024h Post Resize Vertical Residual Register 1 PRZ_VRES1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESIDUAL**															
Type	R/W															

The register specifies vertical residual. It is obtained by PRZ_SRCSZ.HS % PRZ_TARSZ.HT. The allowable maximum value is 4094.

5.24.5.11 Post Resize Block Coarse Shrinking Configuration Register

PRZ+0030h Post Resize Block Coarse Shrinking Configuration Register PRZ_BLKCSCFG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**INTEN**
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VV**		**HV**		**VU**		**HU**		**VY**		**HY**				**CSF**	
Type	R/W		R/W		R/W		R/W		R/W		R/W				R/W	
Reset	00		00		00		00		00		00				00	

The register is for various configuration of Block Coarse Shrinking in Post Resize. Block Coarse Shrinking is dedicated for JPEG decoder. Therefore all processes are based on blocks composed of 8x8 pixels. **Note that all parameters must be set before writing ‘1’ to the register bit PRZ_CON.BLKCSENA**.

CSF It stands for Coarse Shrink Factor. The value specifies the scale factor in coarse shrink pass.
- **00** Image size does not change after coarse shrink pass.
- **01** Image size becomes 1/4 of original size after coarse shrink pass.
- **10** Image size becomes 1/16 of original size after coarse shrink pass.
- **11** Image size becomes 1/64 of original size after coarse shrink pass.

HY Horizontal sampling factor for Y-component
- **00** Horizontal sampling factor for Y-component is 1.
- **01** Horizontal sampling factor for Y-component is 2.
- **10** Horizontal sampling factor for Y-component is 4.
- **11** No Y-component.

VY Vertical sampling factor for Y-component

00 Vertical sampling factor for Y-component is 1.
01 Vertical sampling factor for Y-component is 2.
10 Vertical sampling factor for Y-component is 4.
11 No Y-component.

HU Horizontal sampling factor for U-component
00 Horizontal sampling factor for U-component is 1.
01 Horizontal sampling factor for U-component is 2.
10 Horizontal sampling factor for U-component is 4.
11 No U-component.

VU Vertical sampling factor for U-component
00 Vertical sampling factor for U-component is 1.
01 Vertical sampling factor for U-component is 2.
10 Vertical sampling factor for U-component is 4.
11 No U-component.

HV Horizontal sampling factor for V-component
00 Horizontal sampling factor for V-component is 1.
01 Horizontal sampling factor for V-component is 2.
10 Horizontal sampling factor for V-component is 4.
11 No V-component.

VV Vertical sampling factor for V-component
00 Vertical sampling factor for V-component is 1.
01 Vertical sampling factor for V-component is 2.
10 Vertical sampling factor for V-component is 4.
11 No V-component.

INTEN Interrupt Enable. When interrupt for BLKCS is enabled, interrupt will arise whenever BLKCS finishes.
0 Interrupt for BLKCS is disabled.
1 Interrupt for BLKCS is enabled.

5.24.5.12 Post Resize Y-Component Line Buffer Memory Base Address Register

PRZ+0034h **Post Resize Y-Component Line Buffer Memory Base Address Register** **PRZ_YLMBASE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YLMBASE [31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	YLMBASE [15:0]															
Type	R/W															

The register specifies the base address of line buffer for Y-component. It must be 4 bytes-aligned. The valid address is 0x4002_0000 to 0x4004_3fff. It is only useful in block-based mode.

© 2009 MediaTek Inc.

5.24.5.13 Post Resize U-Component Line Buffer Memory Base Address Register

PRZ+0038h **Post Resize U-Component Line Buffer Memory Base Address Register** **PRZ_ULMBASE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ULMBASE [31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ULMBASE [15:0]															
Type	R/W															

The register specifies the base address of line buffer for U-component. The valid address is 0x4002_0000 to 0x4004_3fff. It is only useful in block-based mode.

5.24.5.14 Post Resize V-Component Line Buffer Memory Base Address Register

PRZ+003Ch **Post Resize V-Component Line Buffer Memory Base Address Register** **PRZ_VLMBASE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VLMBASE [31:16]															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLMBASE [15:0]															
Type	R/W															

The register specifies the base address of line buffer for V-component. The valid address is 0x4002_0000 to 0x4004_3fff. It is only useful in block-based mode.

5.24.5.15 Post Resize Fine Resizing Configuration Register

PRZ+0040h **Post Resize Fine Resizing Configuration Register** **PRZ_FRCFG**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WMSZ															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	O_Y2R1	O_IPP1	O_CRZ				PCSF1				VRINTEN	HRINTEN				VRSS
Type	R/W	R/W	R/W				R/W				R/W	R/W				R/W
Reset	0	0	0				00				0	0				0

The register specifies various setting of control for fine resizing, including of horizontal and vertical resizing. **Note that all parameters must be set before horizontal and vertical resizing proceeds.**

VRSS The register bit specifies whether subsampling for vertical resizing is enabled. For throughput issue, vertical resizing may be simplified by subsampling lines vertically. The register bit is only valid in pixel-based mode.

- **0** Subsampling for vertical resizing is disabled.
- **1** Subsampling for vertical resizing is enabled.

HRINTEN HR (Horizontal Resizing) Interrupt Enable. When interrupt for HR is enabled, interrupt will be issued whenever HR finishes.

- **0** Interrupt for HR is disabled.
- **1** Interrupt for HR is enabled.

VRINTEN VR (Vertical Resizing) Interrupt Enable. When interrupt for VR is enabled, interrupt will be issued whenever VR finishes.

- **0** Interrupt for VR is disabled.
- **1** Interrupt for VR is enabled.

PCSF1 Coarse Shrinking Factor 1 for pixel-based resizing. **Only horizontal coarse shrinking is supported for pixel-based resizing**.

- **00** No coarse shrinking.
- **01** Image width becomes 1/2 of original size after coarse shrink pass.
- **10** Image width becomes 1/4 of original size after coarse shrink pass.
- **11** Image width becomes 1/8 of original size after coarse shrink pass.

O_CRZ The register bit is used to select output modules.

- **0** Disable output to CRZ
- **1** Enable output to CRZ

Note: only for source 5 (JPEGDEC)

O_IPP1 The register bit is used to select output modules.

- **0** Disable output to IPP 1
- **1** Enable output to IPP 1

O_Y2R1 The register bit is used to select output modules.

- **0** Disable output to Y2R 1
- **1** Enable output to Y2R 1

Note: If O_CRZ is 1, O_IPP1 and O_Y2R1 are meaningless. If O_CRZ is 0, both could be 1.

WMSZ It stands for Working Memory Size. The register specifies how many lines after horizontal resizing can be filled into working memory. **Its minimum value is 4.**

Note: The total working memory used by PRZ = (PRZ_TARSZ1.WT+ PRZ_TARSZ1.WT%2) * 2 * WMSZ.

5.24.5.16 Post Resize Y Line Buffer Size Register

PRZ+0050h Post Resize Y Line Buffer Size Register PRZ_YLBSIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					YLBSZ											
Type					R/W											

The register specifies line buffer size for image data after coarse shrinking. It is only useful in block-based mode.

YLBSZ It stands for Y-component Line Buffer Size. The register field specifies how many lines of Y-component can be filled into line buffer. Line buffer size for U- and V-component can be determined

according to the sampling factor. For example, if (V_Y, V_U, V_V)=(4,4,2) and line buffer size for Y-component is 32, lines then the line buffer size for U-component is also 32 lines and V-component 16 lines. If line buffer has capacity for whole image after block coarse shrinking, then block coarse shrinking can be used for the application of scaling down by a factor of 2, or 4, or 8. If dual line buffer is used, block coarse shrinking and horizontal resizing can execute parallel. The maximum allowable value is 2048.

1 Line buffer size for Y-component is 1 line.
2 Line buffer size for Y-component is 2 lines.
3 Line buffer size for Y-component is 3 lines.
...

5.24.5.17 Post Resize Pixel-Based Resizing Working Memory Base Address Register

PRZ+005Ch **Post Resize Pixel-Based Resizing Working Memory Base Address Register** **PRZ_PRWMBASE**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PRWMBASE [31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PRWMBASE [15:0]**															
Type	R/W															

The register specifies the base address of working memory in pixel-based resizing mode. It must be 4 bytes-aligned. The valid address is 0x4002_0000 to 0x4004_3fff
When PRZ_CFG.LB_SEL is set, this address should be set as 0x40050000.

5.24.5.18 Post Resize Information Register 0

PRZ+00B0h **Post Resize Information Register 0** **PRZ_INFO0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INFO[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INFO[15:0]**															
Type	RO															

The register shows progress of BLKCS. But they are not real processed width/height. Sampling factors must be taken into consideration.

INFO[31:16] BLKCS y
INFO[15:00] BLKCS x

5.24.5.19 Post Resize Information Register 1

PRZ+00B4 **Post Resize Information Register 1** **PRZ_INFO1**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INFO[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**INFO[15:0]**
Type	RO

The register shows progress of BLK2PEL.

INFO[31:16] BLK2PEL y
INFO[15:00] BLK2PEL x

5.24.5.20 Post Resize Information Register 2

PRZ+00B8 Post Resize Information Register 2 PRZ_INFO2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INFO[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INFO[15:0]**															
Type	RO															

The register shows progress of pixels received from BLKCS in fine resizing stage.

INFO[31:16] Indicate the account of vertical lines received from BLKCS in fine resizing stage.
INFO[15:00] Indicate the account of horizontal pixels received from BLKCS in fine resizing stage. Note that it will become zero when resizing completes.

5.24.5.21 Post Resize Information Register 3

PRZ+00BC Post Resize Information Register 3 PRZ_INFO3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INFO[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INFO[15:0]**															
Type	RO															

The register shows progress of horizontal resizing in fine resizing stage.

INFO[31:16] Indicate the account of horizontal resizing in fine resizing stage in horizontal direction.
INFO[15:00] Indicate the account of horizontal resizing in fine resizing stage in vertical direction.

5.24.5.22 Post Resize Information Register 4

PRZ+00C0 Post Resize Information Register 4 PRZ_INFO4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INFO[31:16]**															
Type	RO															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INFO[15:0]**															
Type	RO															

The register shows progress of vertical resizing in fine resizing stage.

INFO[31:16] Indicate the account of vertical resizing in fine resizing stage in horizontal direction.

 © 2009 MediaTek Inc.

INFO[15:00] Indicate the account of vertical resizing in fine resizing stage in vertical direction.

5.24.6 Application Notes

- Determine line buffer size by taking into consideration of CSF and sampling factor. For example, if CSF=3 and (Vy, Vu, Vv)=(4,x,x) then minimum of line buffer could be 4 instead of 32.
- Working memory. Maximum value is 16 and minimum 4. **Remember that each pixel occupies 2 bytes.** Thus minimum requirement for working memory in pixel-based resizing is (pixel number in a line) x2x4 bytes.
- Configuration procedure for block-based image sources

  PRZ_CFG.PSEL=0;
  PRZ_CFG.PELSRC = **5**;
  PRZ_FRCFG.O_CRZ = **1 or 0;**
  PRZ_FRCFG.O_IPP1 = **1 or 0;**
  PRZ_FRCFG.O_Y2R1 = **1 or 0;**
  PRZ_BLKCSCFG = select CSF, sampling factor, interrupt enable;
  PRZ_YLBBASE = memory base for Y-component;
  PRZ_ULBBASE = memory base for U-component;
  PRZ_VLBBASE = memory base for V-component;
  PRZ_YLBSIZE = line buffer size for Y-component;
  Other same as that for pixel-based image sources
  PRZ_CON = 0x7;
  // Then wait interrupt or polling PRZ_INT.BLKCSINT or PRZ_INT.BLKHRINT or
  // PRZ_INT.BLKVRINT
- Configuration procedure for pixel-based image sources

  PRZ_CFG.PSEL=1;
  PRZ_CFG.PELSRC = **1~4**;
  PRZ_FRCFG.O_IPP1 = **1 or 0;**
  PRZ_FRCFG.O_Y2R1 = **1 or 0;**
  PRZ_SRCSZ = source image size;
  PRZ_TARSZ = target image size;
  PRZ_HRATIO = horizontal ratio;
  PRZ_VRATIO = vertical ratio;
  PRZ_HRES = horizontal residual;
  PRZ_VRES = vertical residual;
  PRZ_FRCFG = working memory size, interrupt enable;
  PRZ_PRWMBASE = working memory base;
  PRZ_CON = 0x6;

  // Then wait interrupt or polling PRZ_INT.PELHRINT or PRZ_INT.PELVRINT

5.25 SPI Interface Controller

5.25.1 General Description

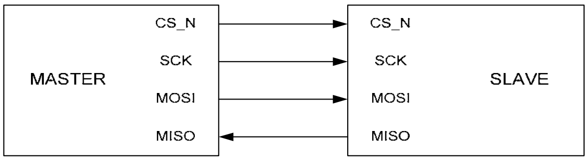


Figure 145: The pin connection between SPI master and SPI slave.

The SPI interface is a bit-serial, four-pin transmission protocol. Figure 145 gives an example of the connection between SPI master and SPI slave. In our chip, the SPI interface controller is a master responsible of the data transmission with the slave.

5.25.1.1 Pin Description

CS_N: The low active chip select signal.
SCK: The (bit) serial clock.
MOSI: The data signal from master output to slave input.
MISO: The data signal from slave output to master input.

5.25.1.2 Transmission Formats

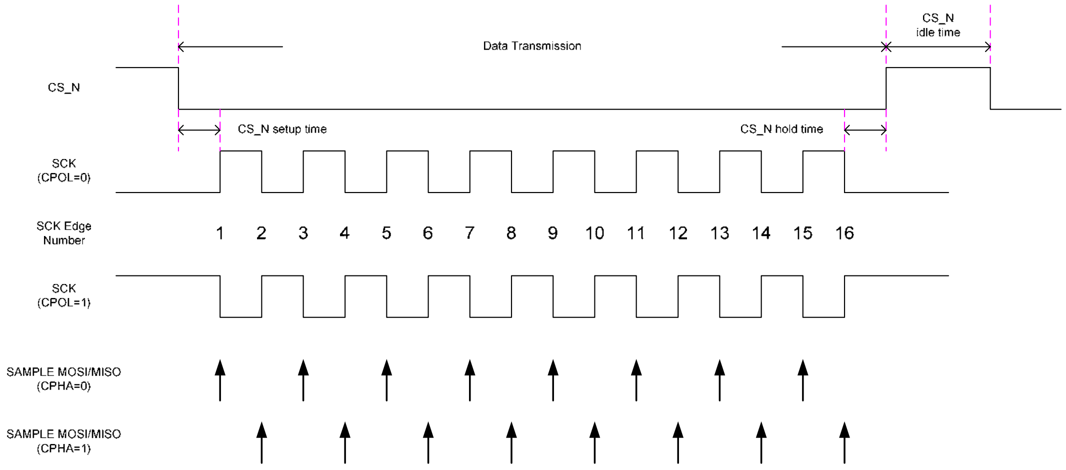


Figure 146: The SPI transmission formats.

Figure 146 gives the waveform during the SPI transmission. The low active CS_N determines the start point and end point of one transaction. The CS_N setup time, hold time, and idle time are also depicted.

The CPOL defines the clock polarity in the transmission. Two kinds of polarity can be adopted, i.e. polarity 0 and polarity 1. Figure 146 gives both of the clock polarity (CPOL) as examples.

The CPHA defines the legal timing to sample MOSI and MISO. Two different ways can be adopted.

5.25.2 Features of the SPI interface controller

The features of the SPI controller (master) are summarized below:

 © 2009 MediaTek Inc.

1. The configurable CS_N setup time, hold time, and idle time.
2. The programmable SCK high time and low time.
3. The configurable transmitting and receiving bit order.
4. Two modes of the source of the data being transmitted can be configured. In TX DMA mode, the SPI controller will automatically fetch the transmitted data (to be put on the MOSI line) from memory. In TX FIFO mode, the data to be transmitted on the MOSI line shall be written to the FIFO before start of the transaction.
5. Two modes of the destination of the data being received can be configured. In RX DMA mode, the SPI controller will automatically store the receiving data (from MISO line) to memory. In RX FIFO mode, the receiving data will just keep in the RX FIFO of the SPI controller. The processor must read back these data by itself.
6. The programmable byte length of transmission.
7. Un-limited length of transmission. This is achieved by the operation of the PAUSE mode. In PAUSE mode, the CS_N signal keeps active (low) after the transmission. At this time, the SPI controller is in PAUSE_IDLE state, ready to receive the resume command. The state transition is shown in Figure 147.

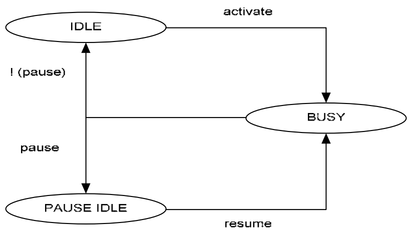


Figure 147: The operation flow with or without PAUSE mode.

5.25.3 The SPI configuration Register

Register Address	Register Function	Acronym
SPI + 0000h	SPI Configuration 0 Register	**SPI_CFG0**
SPI + 0004h	SPI Configuration 1 Register	**SPI_CFG1**
SPI + 0008h	SPI TX Source Address Register	**SPI_TX_SRC**
SPI + 000Ch	SPI RX Destination Address Register	**SPI_RX_DST**
SPI + 0010h	SPI TX DATA FIFO	**SPI_TX_DATA**
SPI + 0014h	SPI RX DATA FIFO	**SPI_RX_DATA**
SPI + 0018h	SPI Command Register	**SPI_CMD**
SPI + 001Ch	SPI Status 0 Register	**SPI_STATUS0**
SPI + 0020h	SPI Status 1 Register	**SPI_STATUS1**

 © 2009 MediaTek Inc.

Table 120 ***The SPI Registers***

5.25.4 Register Definition

SPI+0000h SPI Configuration 0 Register SPI_CFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CS_SETUP_COUNT[7:0]								CS_HOLD_COUNT[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCK_LOW_COUNT[7:0]								SCK_HIGH_COUNT[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CS_SETUP_COUNT[7:0] The chip select setup time = (CS_SETUP_COUNT+1) * CLK_PERIOD, where CLK_PERIOD is the cycle time of the clock the SPI engine adopts.

CS_HOLD_COUNT[7:0] The chip select hold time = (CS_HOLD_COUNT+1) * CLK_PERIOD.

SCK_LOW_COUNT[7:0] The SCK clock low time = (SCK_LOW_COUNT+1) * CLK_PERIOD.

SCK_HIGH_COUNT[7:0] The SCK clock high time = (SCK_HIGH_COUNT+1) * CLK_PERIOD.

SPI+0004h SPI Configuration 1 Register SPI_CFG1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							PACKET_LENGTH[9:0]									
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset							0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PACKET_LOOP_CNT[7:0]								CS_IDLE_COUNT[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PACKET_LENGTH[9:0], PACKET_LOOP_CNT[7:0] The transmission on the SPI bus consists up units bytes. Hence, the PACKET_LENGTH[9:0] define number of bytes in one packet, and PACKET_LOOP_CNT[7:0] defines the number of packets within one transaction. The number of bytes in one packet = PACKET_LENGTH + 1. The number of packets in one transaction = PACKET_LOOP_CNT + 1. Total bytes of one transaction = (PACKET_LENGTH + 1) * (PACKET_LOOP_CNT + 1).

CS_IDLE_COUNT[7:0] The chip select idle time between consecutive transaction = (CS_HOLD_COUNT+1) * CLK_PERIOD.

SPI+0008h SPI TX Source Address Register SPI_TX_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SPI_TX_SRC[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPI_TX_SRC[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

© 2009 MediaTek Inc.

SPI_TX_SRC[31:0] If TX_DMA_EN is set, the data to be putted on the MOSI line are kept in memory in advance, and the SPI controller will automatically read the data from memory. The SPI_TX_SRC define the memory address from which SPI controller starts to read data.

SPI+000Ch SPI RX Destination Address Register SPI_RX_DST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SPI_RX_DST[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPI_RX_DST[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPI_RX_DST[31:0] If RX_DMA_EN is set, the received data from the MISO line will be move to memory automatically by the SPI controller. The SPI_RX_DST define the memory address to which the SPI controller starts to store the data.

SPI+0010h SPI TX DATA FIFO SPI_TX_DATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SPI_TX_DATA[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPI_TX_DATA[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPI_TX_DATA[31:0] The depth of the TX FIFO is 32-bytes. Write to this register will write 4 bytes to the TX FIFO. The TX FIFO pointer will automatically move toward the next four bytes. Read from this register will read 4 bytes from the FIFO, and the TX FIFO pointer automatically moves toward the next four bytes.

SPI+0014h SPI RX DATA FIFO SPI_RX_DATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SPI_RX_DATA[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPI_RX_DATA[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPI_RX_DATA[31:0] The depth of the RX FIFO is 32-bytes. Read from this register will read 4 bytes from the RX FIFO. The RX FIFO pointer will automatically move toward the next four bytes. Write to this register will write 4 bytes to the FIFO, and the RX FIFO pointer automatically moves toward the next four bytes.

SPI_0018h SPI Command Register SPI_CMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															PAUSE_IE	FINISH_IE
Type															R/W	R/W
Reset															0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name			RXMSBF	TXMSBF	TX_DMA_EN	RX_DMA_EN	CPOL	CPHA				PAUSE_EN		RESET	RESUME	CMD_ACT
Type			R/W	R/W	R/W	R/W	R/W	R/W				R/W		WO	WO	WO
Reset			0	0	0	0	0	0				0		0	0	0

PAUSE_IE The interrupt enable bit of pause flag in SPI Status register.

FINISH_IE The interrupt enable bit of finish flag in SPI Status register.

RXMSBF Indicate the data received from MISO line is MSB first or not. Set RXMSBF to 1 for MSB first, otherwise set it to 0.

TXMSBF Indicate the data sent on MOSI line is MSB first or not. Set TXMSBF to 1 for MSB first, otherwise set it to 0.

TX_DMA_EN This bit is the DMA mode enable bit of the data to be transmitted. Default (0) is not to enable.

RX_DMA_EN This bit is the DMA mode enable bit of the data being received.

CPOL This bit is the control bit of the SCK polarity. 0 is CPOL = 0, 1 is CPOL = 1.

CPHA This bit defines the SPI Clock Format 0 or SPI Clock Format 1 during transmission. 0 is CPHA = 0, 1 is CPHA = 1.

PAUSE_EN This bit is the enable bit of the pause mode. Set 1 to enable this mode.

RESET The software reset bit.

RESUME This bit is used when controller is in PAUSE IDLE state. Write 1 to this bit triggers the SPI controller resume transfer from PAUSE IDLE state.

CMD_ACT The command activate bit. Write 1 to this bit triggers the SPI controller starts the transaction.

SPI_001Ch SPI Status 0 Register SPI_STATUS0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name															PAUSE	FINISH
Type															RC	RC
Reset															0	0

FINISH The interrupt status bit in non-pause mode. It will be set by the SPI controller when it completes the transaction, entering the IDLE state.

PAUSE The interrupt status bit in pause mode. It will be set by the SPI controller when it completes the transaction, entering the PAUSE IDLE state.

SPI_0020h SPI Status 1 Register SPI_STATUS1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name																BUSY
Type																RO
Reset																

BUSY This status flag reflects the SPI controller is busy or not. This bit is low active, i.e. 0 represents the SPI controller is busy now.

5.26 TV Controller

5.26.1 General Description

The TV output design can support NTSC/PAL interlaced TV format. The display function includes two components: a TV controller and a TV encoder. The main functions of the TV controller are as follows:

1. Fetch the TV frame buffer.
   - In video playback mode, the source is from the video codec buffer in YUV420 format. In this mode, the TV controller and MPEG4 or H.264 decoder can also communicate to achieve the best performance. see ***Fig. 25.***
   - In image playback mode, the source is in RGB565 format. In this mode, still images can be displayed. The LCM controller can direct the image path to the TV controller. When the LCM controller sends frames to the frame buffer as it does for the LCD display, the TV controller retrieves the frames for display. see ***Fig. 25.***
2. Scale the frame size to fit the TV size. This TV design adopts bilinear interpolation in both horizontal and vertical dimension to scale up the frame. The user can adjust both the location and the size to achieve a suitable appearance.

In NTSC mode, the ideal display area is 720(W) x 480(H), but the actual display area depends on the TV set. Some boundary area may be invisible.
In PAL mode, the ideal display area is 720(W) x 576(H); the actual display area also depends on the TV set.
TV frame updates consume a lot of bandwidth. For interlaced system, one frame contains 2 fields. In NTSC mode, the field update rate is 59.94 frames per second (fps); the field update rate in PAL mode is 50 fps. Performance is bound by the size of the source image. The larger the image size, the higher the bandwidth required to support the TV display.
The controller supports an arbitrary image size up to 640 pixels in height and 480 pixels in width. ***Fig. 29***. depicts the block diagram of the TV controller.

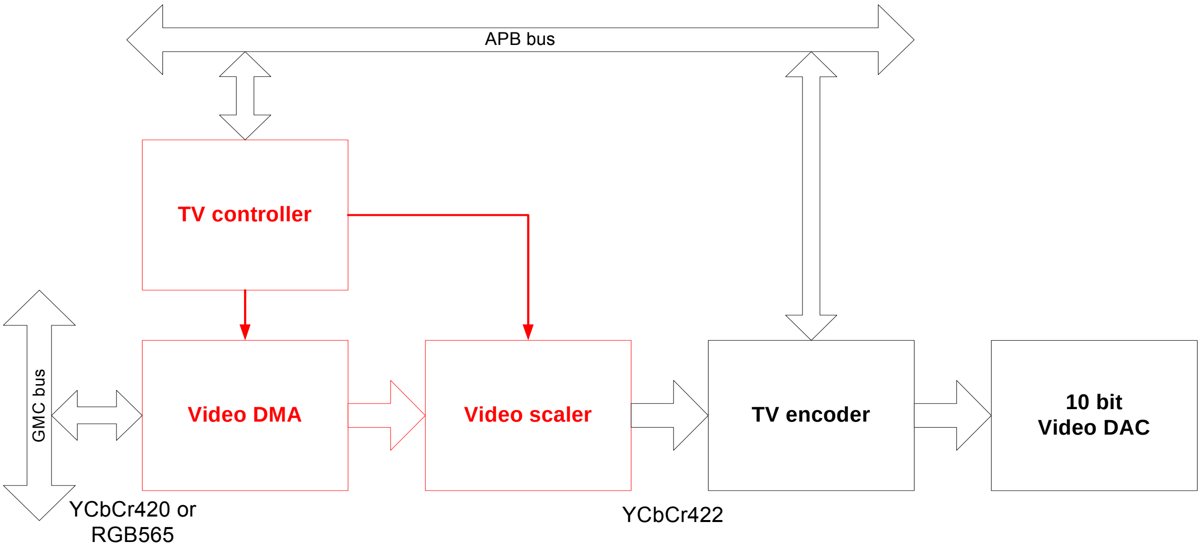


Fig. 29 Block Diagram of the TV Encoder

5.26.2 Pre-Fetch DMA Description

The dedicated and specified pre-fetch DMA for TV is implemented to improve the critical bandwidth problem of TV. With enabling pre-fetch DMA, TV can use the shared internal memory as TV's line buffer. The number of buffer can be programmed but must be the multiple of 4 lines. In order to meet 4-beat burst access, the additional **16 dummy bytes memory size must be kept** for every component pr-fetch buffer for burst mode memory access and vertical interpolation must be enabled. The main functions of TV pre-fetch DMA are as shown in ***Fig. 30***.

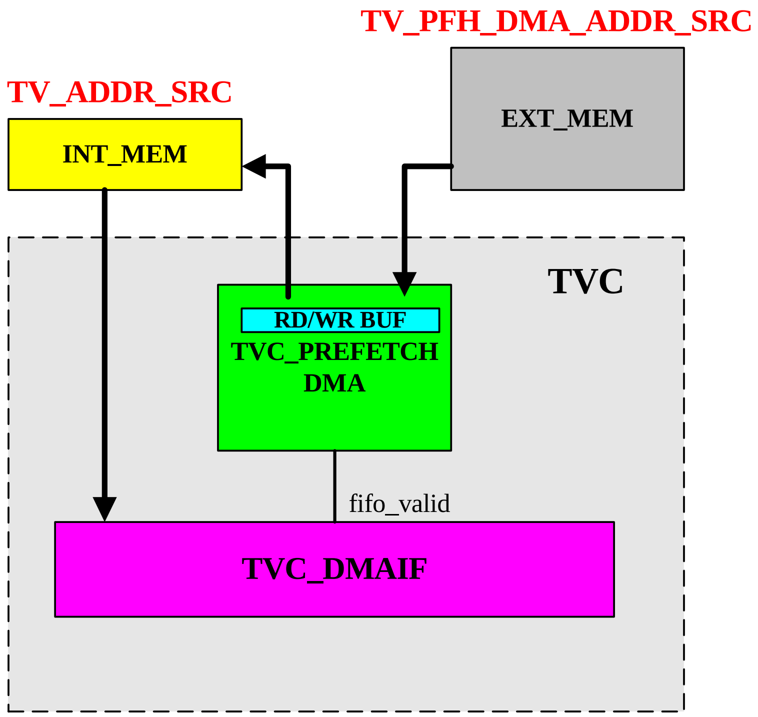


Fig. 30 Pre-Fetch DMA block diagram with TVC.

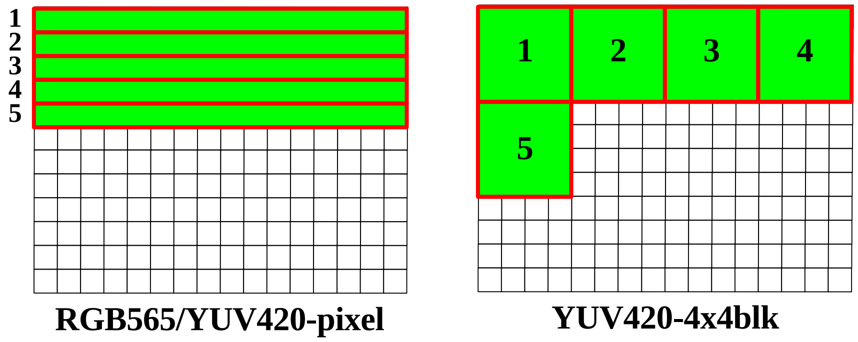


Fig. 25 (a) RGB565/YUV420 sequential format, (b) YUV420 4x4-block format.

5.26.3 Special Register Setting

In order to simplify the address generation of TV controller, some special setting for the address generation needed to be programmed before enabling TV design. The line offset defines the offset byte size from the starting point of every line. The TV design will base on this setting and starting address to decide how to jump to the staring address of the next line, as shown in ***Fig. 32.***

The start/stop pint register can set the final display range in TV device. With the different range setting, the partial or complete video appearance will be displayed in TV device, as shown in ***Fig. 33.***

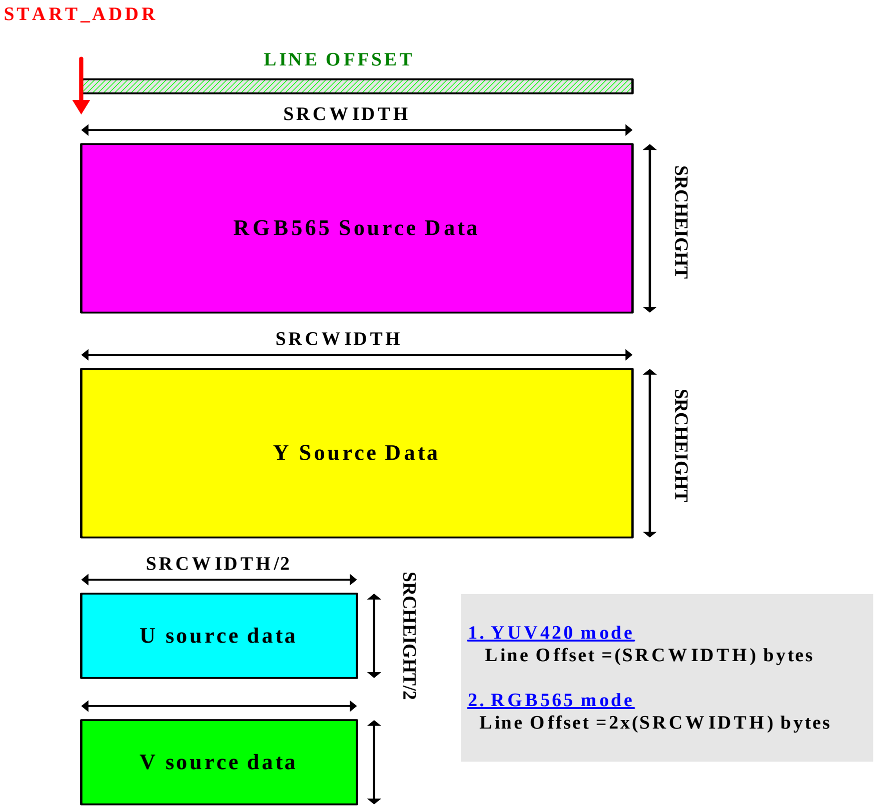


Fig. 32 The line Offset setting in TV address generation.

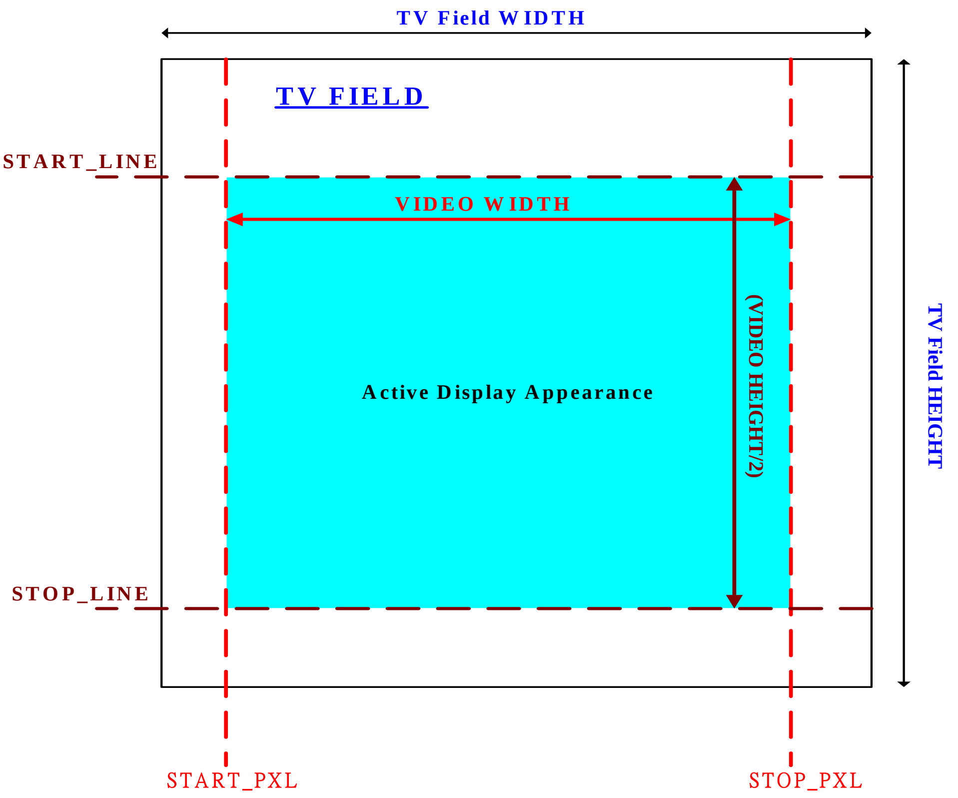


Fig. 33 Display Range Setting.

5.26.4 Register Definitions

Register Address	Register Function	Acronym
TVC + 0000h	TVC enable register	**TVC_ENA**
TVC + 0004h	TVC reset control register	**TVC_RSTB**
TVC + 0008h	TVC control register	**TVC_CON**
TVC + 000Ch	TVC Y data source address register	**TVC_YADR_SRC**
TVC + 0010h	TVC U data source address register	**TVC_UADR_SRC**
TVC + 0014h	TVC V data source address register	**TVC_VADR_SRC**
TVC + 0018h	TVC data source line offset register	**TVC_LINE_OFFSET**
TVC + 001Ch	TVC horizontal scaling coefficient register	**TVC_HCOEF**
TVC + 0020h	TVC vertical scaling coefficient register	**TVC_VCOEF**
TVC + 0024h	TVC frame source width control register	**TVC_SRCWIDTH**
TVC + 0028h	TVC frame source height control register	**TVC_SRCHEIGHT**
TVC + 002Ch	TVC frame target width control register	**TVC_TARWIDTH**
TVC + 0030h	TVC frame target height control register	**TVC_TARHEIGHT**
TVC + 0034h	TVC start point control register 0-3 register	**TVC_START_POINT**
TVC + 0038h	TVC stop point control register	**TVC_STOP_POINT**

TVC + 003Ch	TVC pre-fetch DMA Y data source address	**TVC_PFH_DMA_YADR_SRC**
TVC + 0040h	TVC pre-fetch DMA U data source address	**TVC_PFH_DMA_UADR_SRC**
TVC + 0044h	TVC pre-fetch DMA V data source address	**TVC_PFH_DMA_VADR_SRC**
TVC + 0048h	TVC pre-fetch DMA data source line offset register	**TVC_PFH_DMA_ LINE_OFFSET**
TVC + 004Ch	TVC pre-fetch DMA FIFO length register	**TVC_PFH_DMA _FIFO_LENGTH**
TVC + 0050h	TVC check line register register	**TVC_CHECK_POINT**
TVC + 0054h	TVC register update control register	**TVC_REG_RDY**
TVC + 0058h	TVC vertical average coefficient register	**TVC_AVG_COEFF**
TVC + 005Ch	TVC background color register	**TVC_BACKGROUND_COLOR**
TVC + 0060h	TVC slow control register	**TVC_SLOW_CTRL**
TVC + 0070h	TVC status register	**TVC_SATUS**
TVC + 0074h	TVC Over-run status register	**TVC_OVR_SATUS**
TVC + 0080h	TVC current read data value register	**TVC_RD_DATA_STS**
TVC + 0084h	TVC current read address register	**TVC_RD_ADDR_STS**
TVC + 0088h	TVC DAC accumulator value register	**TVC_DAC_ACC_VALUE**

Table 121 TV controller Register

TVC+0000h TVC enable register TVC_ENA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**TVEN**
Type																R/W
Reset																0

The register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.

TVEN TV controller enable.

0 Disable the TV frame update and display.

1 Enable the TV frame update and display.

TVC+0004h TVC reset control register TVC_RSTB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RSTB**
Type																R/W
Reset																1

The register is used to reset both the TV controller and the TV encoder and **needs to re-write to 1 if finishing reset**. This control bit is read/write.

RSTB Low-Active reset control bit.

TVC+0008h TVC control register TVC_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHK_EN	OVR_EN			CLIP	AVG	AB_WR	AB_RD	PFH	TV_BUSY	BURST	HIGH_PRI		NOIP	DATA_FMT	
Type	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	
Reset	0	0			0	0	0	0	0	0	0	0		0	0	

This register contains double buffer control, burst-mode control, buffer configuration, vertical interpolation control, and color-space control.

CHK_EN Check Line Interrupt Enable. (when TV_line hit the check line setting, the interrupt is asserted.)

CVR_EN Line buffer Over Run Interrupt Enable. (when TV line buffer is over-run, the interrupt is asserted.)

CLIP Clip mode enable for over-run condition. (in buffer-over-run condition, TV will read less data.)

AVG Average mode Enable. If no vertical resizing, enable this bit can improve flicker but need to suitable average coefficients.

AB_WR Double buffer access control.

- **0** Write write-buffer. The active-buffer will be written at the specified time by hardware.
- **1** Write both write-buffer and active-buffer. The active-buffer and the write-buffer will be written directly.

AB_RD Double buffer access control.

- **0** Read write-buffer.
- 1 Read active-buffer.

PFH Pre-fetch access mode Enable. Please reference ***Fig. 30***. Double buffer.

- 0 Disable PFH DMA mode.
- 1 Enable PFH DMA mode(vertical interpolation must be enabled).

TV_BUSY Enable frame sync handshaking between TV and video codec.

- 0 Disable.
- 1 Enable.

BURST Burst mode for memory read access enable. In order to gain better performance, burst mode is necessary.

- 0 Disable
- 1 Enable.

HIGH_PRI Memory access high priority enable.

- 0 Disable
- 1 Enable.

NOIP Bypass vertical interpolation. Enabling this bit reduces the average data access bandwidth by 2. Double buffer.

- **0** Enable vertical interpolation.
- **1** Bypass vertical interpolation.

DATA_FMT TV source data format. Double buffer.

- **0** RGB565. For LCD frame buffer.
- 1 Reserved.
- 2 YUV420 sequential mode.

 © 2009 MediaTek Inc.

3 YUV420 4x4-block mode.

TVC+000Ch TVC Y data source address register TVC_YADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SRC_Y [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC_Y [15:2]**															
Type	R/W														RO	RO
Reset	0														0	0

SRC_Y Address of RGB565 or Y source data.

This register is a double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the Y source address and must be ***word-align address***. In RGB mode, the register represents the RGB source address.

TVC+0010h TVC U data source address register TVC_UADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SRC_U [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC_U [15:2]**															
Type	R/W														RO	RO
Reset	0														0	0

SRC_U Address of U source data.

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the U source address and must be ***word-align address***. In RGB565 mode, this register has no function.

TVC+0014h TVC V data source address register TVC_VADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SRC_V [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SRC_V [15:2]**															
Type	R/W														RO	RO
Reset	0														0	0

SRC_V Address of V source data.

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the V source address and must be ***word-align address***. In RGB565 mode, this register has no function.

 © 2009 MediaTek Inc.

TVC+0018h TVC data source line offset register TVC_LINE_OFFSET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													TV_LINE_BUF_LIMIT [3:0]			
Type													R/W			
Reset													4'h0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TVC_LINEOFFSET[11:2]											
Type					R/W										RO	RO
Reset					0										0	0

TVC_LINEOFFSET This register is double buffer. This register can define the **byte size (but must be multiple of 4)** of one source line in the memory without respecting to the source width of video. Normally, the line offset is $(2 \times sourcewidth)$ for RGB565 mode or $(sourcewidth)$ for YUV420 mode. In YUV420 mode, the U/V line offset will be half of the programmed line offset. **In pre-fetch mode, it can limit the pre-fetch line buffer size.**

TV_LINE_BUF_LIMIT This register can define the bus priority condition. When the number of TV line buffer that had been filled is not larger than TV_LINE_BUF_LIMIT, the high-priority request will be send to bus and then the request of TV can set the top priority to avoid the buffer-under-run in TV(**BUS_PRI need to be enable**).

TVC+001Ch TVC horizontal scaling coefficient register TVC_HCOEF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							HCOEFY									
Type							R/W									
Reset							10'd222									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							HCOEFX									
Type							R/W									
Reset							10'd62									

HCOEFFX horizontal scaling factor ***(X)***.

HCOEFFY residue value of horizontal scaling ***(Y).***

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.

The scaling coefficients should follow the formula: **(only support enlarging function)**

$$\frac{Ws\,(source\ width)-1}{Wt\,(target\ width)-1}=\frac{X+\frac{Y}{Wt-1}}{256},$$

where X, Y are positive integers, and $0 < Y < W_t - 1$.

For example, if the user needs to scale the image width from 640 pixels to 720 pixels, the formula is:

$$\frac{639}{719}=\frac{227+\frac{371}{719}}{256},$$

values X=227 and Y=371 should be programmed.

TVC+0020h TVC vertical scaling coefficient register TVC_VCOEF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							VCOEFY									
Type							R/W									
Reset							10'd204									
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name							VCOEFX									
Type							R/W									
Reset							10'd76									

VCOEFFX vertical scaling factor ***(X)***.

VCOEFFY residue value of vertical scaling ***(Y).***

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.

The scaling coefficients should follow the formula: **(only support enlarging function)**

$$\frac{Hs\,(source\ height)-1}{Ht\,(t\arg etn\ height)-1} = \frac{X+\frac{Y}{Ht-1}}{256},\ X,Y \in positive\ \ in\,teger$$

TVC+0024h TVC frame source width control register TVC_SRCWIDTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							SRCWIDTH									
Type							R/W									
Reset							10'd176									

SRCWIDTH Image/Video source width.

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. **In YUV mode, the source width must be the multiple of 16; in RGB565 mode, the source width must be a multiple of 2 and less than 720.**

TVC+0028 TVC frame source height control register TVC_SRCHEIGHT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							SRCHEIGHT									
Type							R/W									
Reset							10'd144									

SRCHEIGHT Image/Video source height.

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.

In order to meet NTSC/PAL interlaced TV system, the source height must be the multiple of 2 and less than 480(NTSC) or 576(PAL).

TVC+002C TVC frame target width control register TVC_TARWIDTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name							TARWIDTH
Type							R/W
Reset							10'd640

TARWIDTH Image/Video target width.
This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. **In order to meet NTSC/PAL interlaced and YCbCr422 TV system, the target width must be the multiple of 2 and less than 720.**

TVC+0030 TVC frame target height control register TVC_TARHEIGHT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							TARHEIGHT									
Type							R/W									
Reset							10'd480									

TARHEIGHT Image/Video target height.
This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.
In order to meet NTSC/PAL interlaced TV system, the target width must be less than 480(NTSC) or 576(PAL).

TVC+0034h TVC start point control register TVC_START_POINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							START_PXL									
Type							R/W									
Reset							10'd0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							START_LINE									
Type							R/W									
Reset							10'd21									

This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect.
This register is used to control the position of frame displayed on TV. Setting START_PXL to 0 and START_LINE to 21(NTSC) or 22(PAL) aligns the frame to the top-left corner of display. Please reference *Fig. 33*.

START_PXL Starting pixel position in a line.
START_LINE Starting line of display.

TVC+0038h TVC stop point control register TVC_STOP_POINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							STOP_PXL									
Type							R/W									
Reset							0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name								**STOP_LINE**								
Type								R/W								
Reset								0								

This register is to define the number of displayed pixels and lines every field in TV. At the start of each frame, the enable bit is latched into the active buffer and takes effect. Please reference *Fig. 33*.
This register is used to control the display range of field displayed on TV. Setting STOP_PXL to the value of (target width + start_pixel) and STOP_LINE to the ((target_height/2) + start_line) can display the complete frame of video.

STOP_PXL Starting pixel position in a line.
STOP_LINE Starting line of display field.

TVC+003Ch TVC pre-fetch DMA Y data source address TVC_PFH_DMA_YADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PFH_DMA_SRC_Y [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PFH_DMA_SRC_Y [15:2]**															
Type	R/W														RO	RO
Reset	0														0	0

PFH_DMA_SRC_Y Address of Pre-fetch DMA's line buffer for RGB565/Y data.
This register is a double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the Y source address and ***must be in internal memory and word-align address***. In RGB mode, the register represents the RGB source address. Please reference ***Fig. 30.***

TVC+0040h TVC pre-fetch DMA U data source address TVC_PFH_DMA_UADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PFH_DMA_SRC_U [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PFH_DMA_SRC_U [15:2]**															
Type	R/W														RO	RO
Reset	0														0	0

PFH_DMA_SRC_U Address of Pre-fetch DMA's line buffer for U data.
This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the U source address and ***must be in internal memory and word-align address***. In RGB mode, this register has no function. Please reference *Fig. 30*.

TVC+0044h TVC pre-fetch DMA V data source address TVC_PFH_DMA_VADR_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	PFH_DMA_SRC_V [31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PFH_DMA_SRC_V [15:2]															
Type	R/W														RO	RO
Reset	0														0	0

PFH_DMA_SRC_V Address of Pre-fetch DMA's line buffer for V data.
This register is double buffer. At the start of each frame, the enable bit is latched into the active buffer and takes effect. In YUV mode, the register represents the V source address and ***must be in internal memory and word-align address***. In RGB mode, this register has no function. Please reference *Fig. 30*.

TVC+0048h TVC pre-fetch DMA data source line offset register TVC_PFH_DMA_LINE_OFFSET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TVC_PFH_DMA_LINE_OFFSET[11:2]											
Type					R/W										RO	RO
Reset					0										0	0

TVC_PFH_DMA_LINE_OFFSET This register is double buffer. This register can define the **byte size of one source image's line (the setting must be multiple of 16. If not, extend the byte size setting to the multiple of 16.)** In YUV mode, the U/V line offset will be half of the programmed offset address.

TVC+004Ch TVC pre-fetch DMA FIFO length register TVC_PFH_DMA_FIFO_LENGTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FIFO_LENGTH[6:0]						
Type										R/W						
Reset										0						

FIFO_LENGTH This register is double buffer. This register can define the FIFO length in internal fast memory (define by TVC_Y/U/VADR_SRC). The FIFO length must be the multiple of 4 and minimum number is 8. That means we must define the multiple of 4 lines if we want to enable this function to speed up TV output design.

TVC+0050h TVC check line register TVC_CHECK_POINT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							TV_CHECK_LINE2[9:0]									
Type							R/W									
Reset							10'h480									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name							**TV_CHECK_LINE1[9:0]**
Type							R/W
Reset							10'h160

This register is double buffer. This register can set one checking point. When TV line reaches this point, the interrupt of TVC will be asserted if INT_EN has been set to 1.

TV_CHECK_LINE1 TV check points for Even field
TV_CHECK_LINE2 TV check points for Odd field

TVC+0054h TVC register update control register TVC_REG_RDY

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**REG_RDY**
Type																R/W
Reset																0

This register indicates that the double buffer register data is ready to be latched into active buffer.
At the start of each frame, the hardware monitors the bit. If the bit is set to 1 by the software, the double buffer register is latched into active buffer synchronously, and the REG_RDY bit is automatically cleared.

REG_RDY Double buffer control bit.

TVC+0058h TVC vertical average coefficient register TVC_AVG_COEFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**COEFF2[8:0]**								
Type								R/W								
Reset								0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**COEFF1[8:0]**								
Type								R/W								
Reset								9'h100								

This register can define the average coefficient if neighboring vertical line average is necessary to improve the flicker.

COEFF1 Ratio for the current line(9'h100 is 100%).
COEFF2 Ratio for the neighboring (last) line (9'h0 is 0%)

TVC+005Ch TVC background color register TVC_BACKGROUND_COLOR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**U_COLOR[7:0]**							
Type									R/W							
Reset									8'h80							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**V_COLOR[7:0]**								**Y_COLOR[7:0]**							
Type	R/W								R/W							
Reset	8'h80								8'h0							

Y_COLOR This register defines the Luminance(Y) value of TV's background.
U_COLOR This register defines the Chrominance (U) value of TV's background

V_COLOR This register defines the Chrominance (V) value of TV's background

TVC+0060h TVC slow control register TVC_SLOW_CTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					CLIP_FETCH_NUM[3:0]				CLIP_BUF_HIGH_NUM[3:0]				CLIP_BUF_LOW_NUM[3:0]			
Type					R/W				R/W				R/W			
Reset					4'h2				4'h5				4'h4			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							CLIP_SRCWIDTH[9:0]									
Type							R/W									
Reset							10'd176									

CLIP_SRCWIDT This register defines the clipping source width when tvc's buffer is over-run.
CLIP_BUF_LOW_NUM This register defines the trigged condition for clipping.(tv_buffer_cnt <= this setting)
CLIP_BUF_HIGH_NUM This register defines the disabled condition for clipping.(tv_buffer_cnt > this setting)
CLIP_FETCH_LINE_NUM This register defines the trigged condition for clipping.(tv_fetch_line <= this setting)

TVC+0070h TVC status register TVC_SATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CHK_IRQ	OVR_IRQ	CLIP	H_PRI	TV_BUSY	FIELD	TV_LINE									
Type	R/W	R/W	RO	RO	RO	RO	RO									
Reset	1	1	0	0	0	0	10'h0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TVC_DMAIF_STATE				HS_BUF_CNT			TV_LB_CNT			
Type						RO				RO			RO			
Reset						3'h0				3'h0			4'h0			

TV_LB_CNT Number of Line buffer that has been fetch data in(max number is 8).
HS_BUF_CNT Number of Working buffer that has been fetch data in(max number is 4).
TVC_DMAIF_STATE TVC DMA interface state machine

0 IDLE
1 ODD_FIELD_INIT
2 CHECK_LINE_REQ
3 INIT_VSCA_PTR
4 INIT_HS
5 WAIT_EMPTY
6 WAIT_LINE_OK

TV_LINE Index of TV line that displays now.
FIELD Index of TV field that displays now(0:Even field; 1: Odd field).
TV_BUSY TV busy has been asserted and stalls video decoding.
H_PRI High primary bus access request has been asserted.
CLIP Clip function is working.
OVR_IRQ TVC over-run interrupt status report. If TVC's interrupt has been asserted, the status will be set to 0. After reading this status, it must be set to 1.

CHK_IRQ TVC check line interrupt status report. If TVC's interrupt has been asserted, the status will be set to 0. After reading this status, it must be set to 1.

TVC+0074h TVC Over-run status register TVC_OVR_SATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**OVR_LINE_CNT [9:0]**									
Type							RO									
Reset							10'h0									

OVR_LINE_CNT Number of Line over-run in one field.

TVC+0080h TVC current read data status register TVC_RD_DATA_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RD_DATA [31:16]**															
Type	RO															
Reset	16'h0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RD_DATA [15:0]**															
Type	RO															
Reset	16'h0															

RD_DATA Value of TV read data from TV source buffer (**TVC_YADR_SRC, TVC_UADR_SRC, TVC_UADR_SRC**).

TVC+0084h TVC current read address status register TVC_RD_ADDR_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RD_ADDR [31:16]**															
Type	RO															
Reset	16'h0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RD_ADDR [15:0]**															
Type	RO															
Reset	16'h0															

RD_ADDR Current address that is read by TV (**TVC_YADR_SRC, TVC_UADR_SRC, TVC_UADR_SRC**).

TVC+0088h TVC DAC accumulator value register TVC_DAC_ACC_VALUE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TVC_DAC_ACC_VALUE [31:16]**															
Type	RO															
Reset	16'h0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TVC_DAC_ACC_VALUE [15:0]**															
Type	RO															
Reset	16'h0															

TVC_DAC_ACC_VALUE Accumulated DAC output value until TV Line = Check line

5.27 TV encoder

5.27.1 General Description

TV encoder receives a YCbCr stream from the video scaler and encodes the stream into NTSC/PAL signal. *Fig. 34* shows the block diagram of the TV encoder.

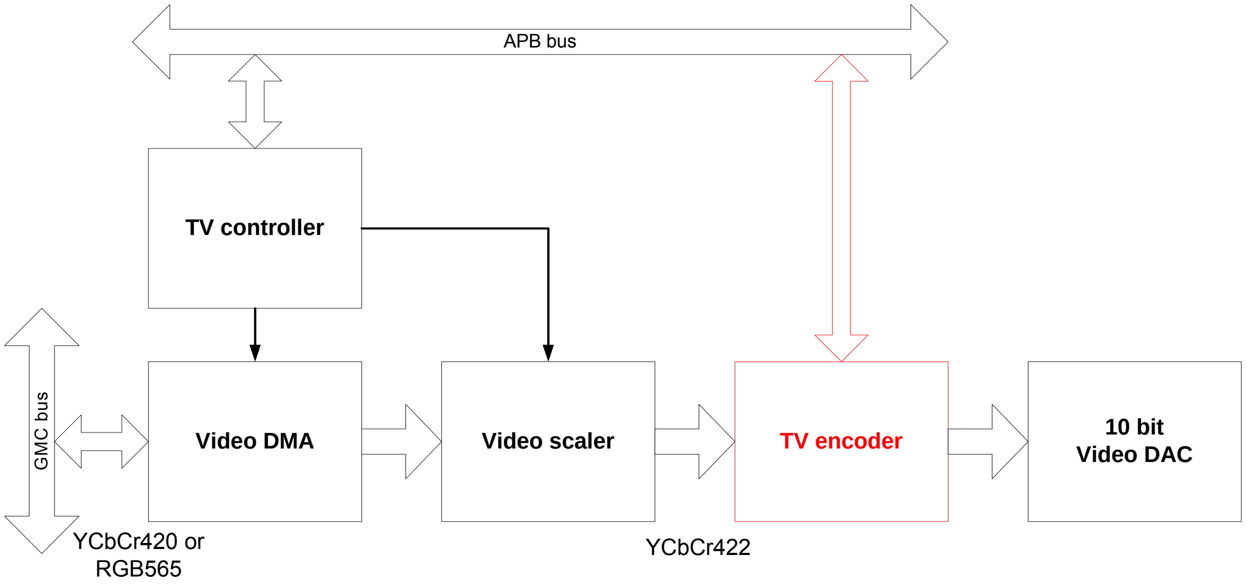


Fig. 34 Block Diagram of TV Encoder

5.27.2 Register Definitions

Register Address	Register Function	Acronym
TVE + 0000h	TV encoder encode mode control register	**TVE_MODE**
TVE + 0004h	TV encoder scale control register	**TVE_CSCALE**
TVE + 0008h	TV encoder DAC control register	**TVE_DACTRL**
TVE + 000Ch	TV encoder Burst level control register	**TVE_BURST**
TVE + 0010h	TV encoder color frequency control register	**TVE_FREQ**
TVE + 0014h	TV encoder Slew control register	**TVE_SLEW**
TVE + 0028h	TV encoder Luma low-pass filter coefficients 10-11 register	**TVE_YLPF**
TVE + 002Ch	TV encoder Luma low-pass filter coefficients 12-15 register	**TVE_YLPFD**
TVE + 0030h	TV encoder Luma low-pass filter coefficients 16-	**TVE_YLPFE**

	19 register	
TVE + 0034h	TV encoder Chrominance low-pass filter coefficients 0-3 register	**TVE_CLPFA**
TVE + 0038h	TV encoder Chrominance low-pass filter coefficients 4-7 register	**TVE_CLPFB**
TVE + 003Ch	TV encoder Chrominance low-pass filter coefficients 8-9 register	**TVE_CLPFC**
TVE + 0040h	TV encoder Gamma correction coefficient 0 register	**TVE_GAMMAA**
TVE + 0044h	TV encoder Gamma correction coefficients 1-2 register	**TVE_GAMMAB**
TVE + 0048h	TV encoder Gamma correction coefficients 3-4 register	**TVE_GAMMAC**
TVE + 004Ch	TV encoder Gamma correction coefficients 5-6 register	**TVE_GAMMAD**
TVE + 0050h	TV encoder Gamma correction coefficients 7-8 register	**TVE_GAMMAE**
TVE + 0054h	TV encoder Burst level control register	**TVE_DACTRL**
TVE + 0060h	TV encoder Software reset control register	**TVE_RSTB**
TVE + 0070h	TV encoder plug-in/out control register	**TVE_PLUG**
TVE + 0074h	TV encoder interrupt enable register	**TVE_INTREN**
TVE + 0078h	TV encoder interrupt status register	**TVE_INTR_STATUS**
TVE + 007Ch	TV encoder plug detection value register	**TVE_PLUG_VALUE**
TVE + 0080h	TV encoder plug detection line timing register	**TVE_PLUG_LINE**
TVE + 0084h	TV encoder plug detection pixel timing register	**TVE_PLUG_PXL**

Table 122 TV encoder Register

TVE+0000h Encoder mode control register TVE_MODE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															TVTYPE	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	UVSWP	BLKER	SLOFF			SYDEL	YDEL		CUPOF	YLPON	CLPON	CLPSEL		SETUP	CBON	ENCON
Type	R/W	R/W	R/W			R/W	R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0			0	0		0	0	0	0		0	0	0

TVTYPE TV type.

- **00** NTSC (525 lines, no phase alternation line)
- **01** PAL-M (525 lines, with phase alternation line)
- **10** PAL-C (625 lines, with phase alternation line)
- **11** PAL (625 lines, with phase alternation line)

UVSWP U/V swap.

BLKER Blacker than black mode on.

SLOFF Slew at the beginning and at the end of the horizontal active area off.
SYDEL Delay of Y (half sample resolution).
YDEL Delay of Y (one sample resolution). (Recommended setting is 2.)
CUPOF Chrominance (chroma) of component up-sample off.
YLPON Luminance (luma) low-pass filter on. (Recommended setting is 1.)
CLPON Chroma low-pass filter on. (Recommended setting is 1.)
CLPSEL Chroma low-pass filter coefficient selection.
SETUP 7.5IRE setup enable. (M) NTSC and (M, N) PAL have a blanking pedestal.
CBON Enable the color bar.
ENCON Enable the TV encoder.

TVE+0004h Scale control register TVE_CSCALE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													**BLANK**			
Type													R/W			
Reset													0x4			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VSCALE**								**USCALE**							
Type	R/W								R/W							
Reset	0x5A (90)								0x5A (90)							

USCALE Scale of U (USCALE/128). $U_{new} = U \cdot USCALE$

VSCALE Scale of V (VSCALE/128). $V_{new} = V \cdot VSCALE$

BLANK Luma data at this level (BLANKx4) is presented as blank. This blacker function must be enabled.

TVE+0008h DAC control register TVE_DACTRL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						**RG_CKB_EN**	**RG_CK_SRC**	**TEST_ANA[3:0]**					**TRIM**			
Type						R/W	R/W	R/W					R/W			
Reset						0	0	0					0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**BLACKER**			**TEST_COMP_EN**	**VPLUGREF**		**PLUG_DET_EN**			**PDN_HAIBIAS**	**PDN_DAC2**	**PDN_DAC1**	**PDN_DAC0**	**PDN_BGREF**	**DAC_EN**
Type		R/W			R/W	R/W		R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset		0			0	0		0			0	0	0	0	0	1

RG_CKB_EN Enable 27MHz inverse clock source
- **0** Disable.
- **1** Enable.

RG_CK_SRC TVDAC clock source select
- **0** TVE Clock used
- **1** TVPLL generated clock

TEST_ANA. Plug-in threshold level selection
TRIM Trimming code for BGVref.
BLACKER BLANK luma data adjustment enable.
- **0** Disable.

1 Enable

TEST_COMP_EN Comparator test enable.

0 Disable.

1 Enable.

VPLUGREF Plug-in detect threshold selection.

PLUG_DET_EN Plug-in/out detect enable. **(must enable plug-in or plug-out interrupt)**

0 Disable.

1 Enable.

PDN_HAIBIAS Half bias current power down mode.

0 Power down.

1 Power up.

PDN_DAC2 DAC current is $I_{DAC} = \frac{7}{8} I_{ref}$ enable control.

0 Power down.

1 Power up.

PDN_DAC1 DAC current is $I_{DAC} = \frac{3}{4} I_{ref}$ enable control.

0 Power down.

1 Power up.

PDN_DAC0 DAC current is $I_{DAC} = \frac{1}{2} I_{ref}$ enable control.

0 Power down.

1 Power up.

PDN_BGREF BGVref power down control.

0 Power down.

1 Power up.

DAC_EN DAC enable.

TVE+000Ch Burst level control register TVE_BURST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**UPQINI**							
Type									R/W							
Reset									0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BRSTLVL**							
Type									R/W							
Reset									0x3A (58)							

UPQINI Phase offset of the color burst.

BRSTLVL Color burst level.

TVE+0010h Color frequency control register TVE_FREQ

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**BFP2**												
Type				R/W												
Reset				0xdd0 (3536)												

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**BFP1**								
Type								R/W								
Reset								0x10f (271)								

Burst frequency control.

$$Burst\ frequency = 27\text{MHz} \times \frac{BFP1 + \frac{BFP2 + \frac{X}{625}}{4H}}{2048},$$

where H is the pixel clock cycle number per line.

Use the following table to get BFP1 and BFP2 (in decimal).

TV type	**H**	**X**	**BFP1**	**BFP2**
NTSC	1716	0	271	3536
PAL	1728	67	336	2061

BFP2 Color burst frequency synthesis value 2.

BFP1 Color burst frequency synthesis value 1.

TVE+0014h Slew control register TVE_SLEW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**SLEWUP**								
Type								R/W								
Reset								0xc8 (200)								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**SLEWDN**										
Type						R/W										
Reset						0x6A4 (1700)										

SLEWUP Begin cycle of valid pixel with slew rate control.

SLEWDN End cycle of valid pixel with slew rate control.

TVE+0028h Luma low-pass filter coefficients 10-11 register TVE_YLPFC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**YLPF11**								**YLPF10**					
Type			R/W								R/W					
Reset			0x32 (-14)								0x2 (2)					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

YLPF11 Luma low-pass filter coefficient 11. Signed integer.

YLPF10 Luma low-pass filter coefficient 10. Signed integer.

TVE+002Ch Luma low-pass filter coefficients 12-15 register TVE_YLPFD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**YLPF15**								**YLPF14**						

Type		R/W								R/W						
Reset		0x2 (2)								0x1e (30)						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**YLPF13**								**YLPF12**					
Type			R/W								R/W					
Reset			0X3d (-3)								0X25 (-27)					

YLPF15 Luma low-pass filter coefficient 15. Signed integer.
YLPF14 Luma low-pass filter coefficient 14. Signed integer.
YLPF13 Luma low-pass filter coefficient 13. Signed integer.
YLPF12 Luma low-pass filter coefficient 12. Signed integer.

TVE+0030h Luma low-pass filter coefficients 16-19 register TVE_YLPFE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**YLPF19**								**YLPF18**							
Type	R/W								R/W							
Reset	0x90 (144)								0xb4 (180)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**YLPF17**								**YLPF16**							
Type	R/W								R/W							
Reset	0Xff (-1)								0Xc7 (-57)							

YLPF19 Luma low-pass filter coefficient 19. Must be unsigned. Hardware extends to 9 bits.
YLPF18 Luma low-pass filter coefficient 18. Must be unsigned. Hardware extends to 9 bits.
YLPF17 Luma low-pass filter coefficient 17. Signed integer.
YLPF16 Luma low-pass filter coefficient 16. Signed integer.

TVE+0034h Chrominance low-pass filter coefficients 0-3 register TVE_CLPFA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CLPF3**								**CLPF2**					
Type			R/W								R/W					
Reset			0x18								0xd					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**CLPF1**								**CLPF0**					
Type			R/W								R/W					
Reset			0x10								0x1					

CLPF3 Chrominance low-pass filter coefficient 3.
CLPF2 Chrominance low-pass filter coefficient 2.
CLPF1 Chrominance low-pass filter coefficient 1.
CLPF0 Chrominance low-pass filter coefficient 0.

TVE+0038h Chrominance low-pass filter coefficients 4-7 register TVE_CLPFB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CLPF7**								**CLPF6**					
Type			R/W								R/W					
Reset			0x25								0x34					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**CLPF5**								**CLPF4**					
Type			R/W								R/W					
Reset			0x20								0x21					

CLPF7 Chrominance low-pass filter coefficient 7.
CLPF6 Chrominance low-pass filter coefficient 6.
CLPF5 Chrominance low-pass filter coefficient 5.
CLPF4 Chrominance low-pass filter coefficient 4.

TVE+003Ch Chrominance low-pass filter coefficients 8-9 register TVE_CLPFC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name			**CLPF9**								**CLPF8**					
Type			R/W								R/W					
Reset			0x27								0x3c					

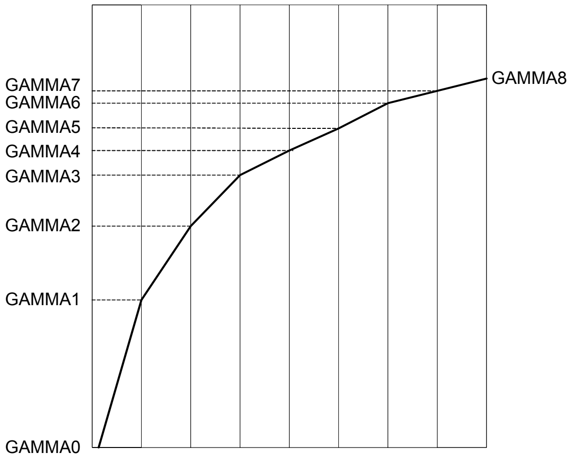
CLPF9 Chrominance low-pass filter coefficient 9.
CLPF8 Chrominance low-pass filter coefficient 8.

TVE+0040h Gamma correction coefficient 0 register TVE_GAMMAA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**GAMMA0**											
Type					R/W											
Reset					0											
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name																
Type																
Reset																

GAMMA0~GAMMA8 indicate the turning points of a piecewise linear approximation for a gamma curve. By default, the values form a perfect linear equation with no gamma correction.

Gamma correction is performed on Luma only.



GAMMA0 Gamma correction coefficient 0.

TVE+0044h Gamma correction coefficients 1-2 register TVE_GAMMAB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					GAMMA2											
Type					R/W											
Reset					0x314											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GAMMA1											
Type					R/W											
Reset					0x18a											

GAMMA2 Gamma correction coefficient 2.

GAMMA1 Gamma correction coefficient 1.

TVE+0048h Gamma correction coefficients 3-4 register TVE_GAMMAC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					GAMMA4											
Type					R/W											
Reset					0x629											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GAMMA3											
Type					R/W											
Reset					0x49e											

GAMMA4 Gamma correction coefficient 4.

GAMMA3 Gamma correction coefficient 3.

TVE+004Ch Gamma correction coefficients 5-6 register TVE_GAMMAD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					GAMMA6											
Type					R/W											
Reset					0x93d											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GAMMA5											
Type					R/W											
Reset					0x7b3											

GAMMA6 Gamma correction coefficient 6.

GAMMA5 Gamma correction coefficient 5.

TVE+0050h Gamma correction coefficients 7-8 register TVE_GAMMAE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					GAMMA8											
Type					R/W											
Reset					0xc52											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GAMMA7											
Type					R/W											
Reset					0xac8											

GAMMA8 Gamma correction coefficient 8.

GAMMA7 Gamma correction coefficient 7.

 © 2009 MediaTek Inc.

TVE+0060h Software reset control register TVE_RSTB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**RSTB**
Type																R/W
Reset																1

RSTB Set to 0 to reset theTV encoder. And set to 1 to finish the reset process

TVE+0070h TV encoder DAC plug-in/out status register TVE_PLUG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**COMPRESS**
Type																RO
Reset																0

COMPRESS TVDAC sense plug-in or plug-out. This bit is valid in the plug-detection period.

0 Plug-Out

1 Plug-in

TVE+0074h TV encoder interrupt enable register TVE_INTREN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**PLUG_OUT_DET**	**PLUG_IN_DET**
Type															R/W	R/W
Reset															0	0

PLUG_OUT_DET Set to 1 to enable plug-out detection process and enable plug-out interrupt. **(plug-in detection must be disabled)**

PLUG_IN_DET Set to 1 to enable plug-in detection process and enable plug-in interrupt. **(plug-out detection must be disabled)**

TVE+0078h TV encoder interrupt status register TVE_INTR_STS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name															PLUG_OUT	PLUG_IN
Type															R/W	R/W
Reset															0	0

PLUG_OUT 1 indicate TV cable is plug-out and will assert TVE's interrupt. After finishing plug-out detection status reading, write 0 in this bit to clear the old status.

PLUG_IN 1 indicate TV cable is plug-in and will assert TVE's interrupt. After finishing plug-in detection status reading, write 0 in this bit to clear the old status.

TVE+007Ch TV encoder plug detection value register TVE_PLUG_VALUE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					PLUG_VALUE[11:0]											
Type					R/W											
Reset					12'hfff											

PLUG_VALUE Digital code that TV encoder will send to TVDAC in plug detection period. Please see ***Fig. 35*.**

TVE+0080h TV encoder plug detection line timing register TVE_PLUG_LINE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							PLUG_LINE[9:0]									
Type							R/W									
Reset							10'd11									

PLUG_LINE Line count is used for plug detection. This line should be selected the dummy line in TV display and we suggest 10'd11. Please see ***Fig. 35*.**

TVE+0084h TV encoder plug detection pixel timing register TVE_PLUG_PXL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						PLUG_END_PXL[10:0]										
Type						R/W										
Reset						11'd426										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						PLUG_START_PXL[10:0]										
Type						R/W										
Reset						11'd326										

PLUG_END_PXL In this ending pixel count, plug detection will finish. (should be >11'd230)

PLUG_START_PXL In this starting pixel count, plug detection will start. (should be >11'd1700) Please see ***Fig. 35*.**

 © 2009 MediaTek Inc.

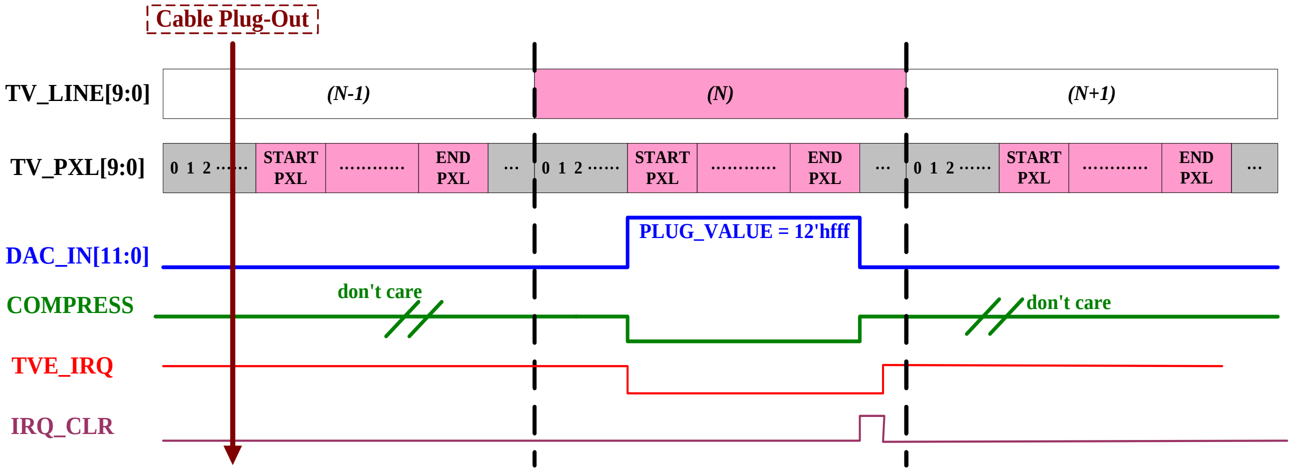


Fig. 35 Plug detection setting.

5.28 Wavetable Synthesizer

5.28.1 General Descriptions

Wavetable synthesizer is an MCU coprocessor which is designed specifically for the feature of fancy sound effects realized by wavetable synthesis. This wavetable synthesizer has several functions including data movement, decompression, interpolation and some mathematic operations, thus the loading of MCU can be released.

In general, this wavetable synthesizer has following functions.

- Data pre-fetch: the wavetable synthesis method, which needs large amount of sample data as well as the data rate, needs an efficient data transfer scheme. A 512-bit data buffer is used as a cache of the fetched data. In this way, a 8-beat burst access in GMC bus can take the advantage of address continuity; some burst mode accesses in external storage device can be activated. Moreover, the optional dynamic switch to 2-beat mode are supported.
- Rewind if needed: wavetable method for ring-tone generation take the advantage of repetition of wave samples. The data transfer unit of this wavetable synthesizer will calculate the sample indices and judge if rewind operation is needed. In brief, it is a ring-buffer scheme plus some additional requirements.
- wavetable format decoder: PCM16, PCM8, a-law, u-law, 4-bit ADPCM and 8-bit ADPCM are supported. If the wave table samples are compressed in order to save storage size and average access count, this wavetable synthesizer has decoders to decompress the 4-bit or 8-bit codes into 16-bit wave samples.
- Interpolation: this wavetable synthesizer is capable of finding the correct phase and the wave samples relative to the current sample index. Using these information, the interpolation unit can generate correct sample data. It reduces data processing time in MCU as well as the data transfer count to system memory.

- Two pole low-pass Filter: this wavetable synthesizer provides an IIR filter with dynamic controlled (programmable) cutoff frequency and resonance. This filter is optional to turn on or off.
- Envelope multiplier: linearly piecewise envelope process is applied to each tone before synthesis. The envelope gain can be programmabled.
- Synthesis: wavetable synthesizer processes the foregoing operations note by note. The processing results are accumulated with other tones and stored in an accumulation memory. Maximum 256 tones can be supported.
- Interrupt the MCU: wavetable synthesizer will interrupt the MCU when all tones are processed. The synthesized results are ready to be read by MCU through APB bus.

A general description of the role played by the wavetable synthesizer can be pictured as in Figure 148. The wavetable synthesizer is controlled by MCU (The registers are defined as in 0). After a trigger signal register is set, the wavetable synthesizer starts to read processing parameters (The working memory are defined as in 5.28.3) of the first note from internal ram through GMC bus. From the processing parameters, the wavetable synthesizer gets the location of the wavetable data and fetches it through GMC bus. After all the wave samples of the current note are processed, the wavetable synthesizer will write some processing parameter values back to the internal ram as the initial values of the next section of wavetable synthesis. Then, the wavetable synthesizer goes on reading the processing parameters of the next note and processes the wave data of the next note. The wavetable synthesizer works until one of the following conditions is met:

(1) The required note number has reached.

(2) MCU sends a stop signal to freeze the transfer (send trigger signal again).

Condition (2) is an abnormal case. It could be happened when MCU resource management fail, including computing time or memory space. In this case, the MCU should abandon current process, lost the tone of moment, and rebuild the data path until the resource is O.K again. To sum up, it is a prevention mechanism for the hardware / MCU interface falls into a dead lock scene.

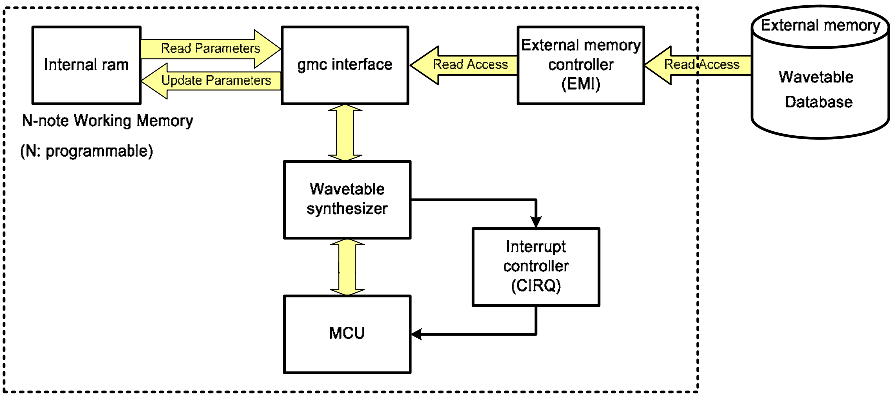


Figure 148 Role played by the wavetable synthesizer.

5.28.2 Register Definitions

The wavetable base address is 0x80090000. Address 0x0 ~ 0xEFF are used for 960 wavetable output data address. If the software read the address 0x80090000/0x800980EFC, the hardware will output the first/second left channel sythesized sample. If the software read the address 4, the hardware will output the first right channel sythesized sample.

Table 123 ***Register table***

Address	Register name	Description
0x80090000	WAVE_DATA_L1	First left channel sythesized sample
0x80090004	WAVE_DATA_R1	First right channel sythesized sample
~	~	~
0x80090EF8	WAVE_DATA_L480	480th left channel sythesized sample
0x80090EFC	WAVE_DATA_R480	480th right channel sythesized sample
0x80090F00	WAVE_TRIG	Wavetable trigger register
0x80090F04	WAVE_SLOW	Wavetable slow down control
0x80090F08	WAVE_PAR_BASE	Parameter working memory base address
0x80090F0C	WAVE_POLLING	Wavetable status polling register
0x80090F10	WAVE_IRQ_ACK	Wavetable interrupt ack register
0x80090F14	WAVE_MAX_VAL	Wave synthesizer maximum value register
0x80090F18	DBG_CUR_NOTE	Debug Register – current note status
0x80090F1C	DBG_CORE_FSM	Debug Register – core state machine
0x80090F20	DBG_PAR_FSM	Debug Register – parameter memory interface state machine
0x80090F24	DBG_DATA_FSM	Debug Register – data memory interface state machine

0x80090F00 Wavetable trigger register WAVE_TRIG

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									NOTE_NUM							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
Reset	0	0	0	0	0	0	0	0	0							

The wavetable synthesizer starts to work when this register is written and stops working when this register is written again before it is finished.

NOTE_NUM This field defines the maximum number of notes to be synthesized.

0x80090F04 Wavetable slow down control WAVE_SLOW

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						BEAT_DYN		WT_SDCNT								SD_DIS
Type	R/W	R/W	R/W	R/W	R/W	R/W		R/W								R/W
Reset	0	0	0	0	0	0		0								1

SD_DIS **0** Enable wavetable slow down mode.
1 Disable wavetable slow down mode.

WT_SDCNT The wait cycles in slow down mode. By setting this field, the hardware will delay **WT_SDCNT** cycles to access the data bus.

BEAT_DYN **0** no dynamic switch to 64bit, 2word burst EMI access when step size is too large.
1 dynamic switch to 64bit, 2word burst EMI access when step size is too large.

0x80090F08 Parameter working memory base address — WAVE_PAR_BASE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PAR_BASE_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PAR_BASE_ADDR[15:0]															
Type	R/W															
Reset	0															

PAR_BASE_ADDR the base address of the parameter working memory.

0x80090F0C Wavetable status polling register — WAVE_POLLING

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																WAVE_BUSY
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE_BUSY Read only. For software to poll the wave synthesizer hardware status.
0 Wavetable synthesizer finish the task and in IDLE state.
1 Wavetable synthesizer is busy.

0x80090F10 Wavetable interrupt ack register — WAVE_IRQ_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																IRQ_ACK

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQ_ACK Software has to write this field as 1 to clear wavetable interrupt source when mcu get the wavetable completed interrupt. Please refer to AP side CIRQ document to get interrupt information.

0x80090F14 Wave synthesizer maximum value register WAVE_MAX_VAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							**MAX_VAL[25:0]**									
Type	R/W	R/W	R/W	R/W	R/W	R/W	RO									
Reset	0	0	0	0	0	0	0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAX_VAL[15:0]**															
Type	RO															
Reset	0															

MAX_VAL Maximum value of 480 output samples

0x80090F18 Debug Register – current note status DBG_CUR_NOTE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**CURR_SAMPLE_COUNT**								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO								
Reset	0	0	0	0	0	0	0	0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**FILTER_EN**	**LOOP_EN**	**CURR_NOTE**							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R0	RO	RO							
Reset	0	0	0	0	0	0	0	0	0							

0x80090F1c Debug Register – core state machine DBG_CORE_FSM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**CORE_FSM_1**				
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO				
Reset	0	0	0	0	0	0	0	0	0	0	0	01h				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**CORE_FSM_0**													
Type	R/W	R/W	RO													
Reset	0	0	0001h													

0x80090F20 Debug Register – parameter memory interface state machine DBG_PAR_FSM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**WPAR_FSM[8:0]**											**RPAR_FSM[17:16]**	
Type	R/W	R/W	R/W	RO									R/W	R/W	RO	
Reset	0	0	0	001h									0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RPAR_FSM[15:0]**															
Type	RO															
Reset	0001h															

0x80090F24 Debug Register – data memory interface state machine DBG_DATA_FSM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**DATA_FSM[17:16]**	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0h	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DATA_FSM[15:0]**															
Type	RO															
Reset	0001h															

5.28.3 Working Memory Definition

The working memory is used to store the processing information of wavetable synthesizer and must be programmed by the software before triggering the hardware (wavetable synthesizer) to work. This working memory occupies a part of internal sysram of which the starting address is 0x40000000. The actual base address of this working memory (a part of) is defined by the software and written to the wavetable synthesizer as WAVE_PAR_BASE.

Table 124 ***Working memory table***

Note number	Base address	Memory name
	WAVE_PAR_BASE +0000h	Note 0 WORK_MEM_0
	≀	≀
Note_0	WAVE_PAR_BASE +0034h	Note 0 WORK_MEM_13
	WAVE_PAR_BASE +0038h	Reverved
	WAVE_PAR_BASE +003ch	Reverved
	WAVE_PAR_BASE +0040h	Note 1 WORK_MEM_0
	≀	≀
Note_1	WAVE_PAR_BASE +0074h	Note 1 WORK_MEM_13
	WAVE_PAR_BASE +0078h	Reverved
	WAVE_PAR_BASE +007ch	Reverved
≀	≀	≀

Note_x+0000h Word 0 in the working memory WORK_MEM_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TOTAL_LEN**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAX_OUT_COUNT**									**TABLE_FORMAT**			**STEREO**	**FILTER_EN**	**LOOP_EN**	**NOTE_ON**
Type	R/W									R/W			R/W	R/W	R/W	R/W

NOTE_ON 0 This note is disabled. Skip to next note.
1 This note is enabled. Keep on wave modulating.

LOOP_EN **0** Looping operation is disabled.
1 Looping operation is enabled.
FILTER_EN **0** IIR filter is turned off.
1 IIR filter is turned on.
STEREO **0** Wavetable is stored as mono format.
1 Wavetable is stored as stereo format.
TABLE_FORMAT Wavetable format
000 16-bit PCM
001 8-bit PCM
010 8-bit A-Law
011 8-bit U-Law
100 8-bit ADPCM
101 4-bit ADPCM
MAX_OUT_COUNT Maximum wavetable output sample number (<=480)
TOTAL_LEN In release mode, if the wavetable sample is larger than TOTAL_LEN, the value will be zero.

Note_x+0004h Word 1 in the working memory WORK_MEM_1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WAVE_BASE[31:16]**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAVE_BASE[15:0]**															
Type	R/W															

WAVE_BASE Base address of the wavetable samples stored in the external memory.

Note_x+0008h Word 2 in the working memory WORK_MEM_2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**REWIND_VAL**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**D2B_LPSTR**															
Type	R/W															

D2B_LPSTR Distance from loop start point to the base address.
REWIND_VAL This field records the loop start point value. If the current sample must be interpolated by the rewind point and the loop start point, the hardware will use this register value instead of fetching the loop start point value in the table.

Note_x+000Ch Word 3 in the working memory WORK_MEM_3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**REWIND_LEN**															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**REWIND_PHASE**															
Type	R/W															

REWIND_LEN The integer part of the distance from the loop start point to the rewind point.

REWIND_PHASE The fractional part of the distance from the loop start point to the rewind point. It is a 6.10 format fractional number (6-bit integer and 10-bit fraction). The absolute address of the rewind point can be described as (WAVE_BASE + D2B_LPSTR + REWIND_LEN + REWIND_PHASE[15:10]).

Note_x+0010h Word 4 in the working memory WORK_MEM_4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ENVELOPE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ENVELOPE_RATE															
Type	R/W															

ENVELOPE Current envelope multipier value. The register should be write back to the working memory after each note of wavetable synthesis is completed. The write-back value should be kept and read-out from memory at next section of the same wavetable syntehsis. It is an unsigned 0.16 format fractional number (0-bit integer and 16-bit fraction).

ENVELOPE_RATE Envelope updated rate. (per output sample). It is a signed -10.25 format fractional number (msb bits are sign value and then $1/2^{10}$, $1/2^{11}$, $1/2^{12}$, etc).

Note_x+0014h Word 5 in the working memory WORK_MEM_5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WAVE_STR_INDEX															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_STR_PHASE															
Type	R/W															

WAVE_STR_PHASE The initial sample phase at each transfer block. The register should be write back to the working memory after each note of wavetable synthesis is completed. The write-back value should be kept and read-out from memory at next section of the same wavetable syntehsis, or phase jitter will occur between successive transfer blocks.

WAVE_STR_INDEX The initial sample index at each transfer block. Like WAVE_STR_PHASE, content of this register should be write back to the working memory after each note of wavetable synthesis is completed. The actual starting sample index is derived by (WAVE_STR_PHASE + WAVE_STR_INDEX).
Note: if (WAVE_STR_PHASE + WAVE_STR_INDEX) exceeds the rewind address, it rewinds by (REWIND_LEN + REWIND_PHASE).

Note_x+0018h Word 6 in the working memory WORK_MEM_6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SHIFT_STEP_RATE															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SHIFT_STEP															
Type	R/W															

© 2009 MediaTek Inc.

SHIFT_STEP The shift step between each sample index. It is a fractional number in 6.10 format. The register should be write back to the working memory after each note of wavetable synthesis is completed.

SHIFT_STEP_RATE The shift step changing rate between each sample index. It is a fractional number in signed -4.19
format (msb bits are sign value and then $1/2^4$, $1/2^5$, $1/2^6$, etc).

ote_x+001Ch Word 7 in the working memory WORK_MEM_7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name										**ADPCM_INI_INDEX**						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADPCM_INI_VAL**															
Type	R/W															

ADPCM_INI_VALUE The initial sample of the ADPCM procedure. After one block of data processing is done, this register write back to working mempry for connection of the next section with the same wave sample.

ADPCM_INI_INDEX The 7-bit ADPCM initial index value. Valid values ranged from 0 to 88. Same as ADPCM_INI_VALUE, this register needs to be written back after one data block processing.

Note_x+0020h Word 8 in the working memory WORK_MEM_8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name										**ADPCM_LP_INDEX**						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ADPCM_LP_VAL**															
Type	R/W															

ADPCM_LP_VALUE The ADPCM sample value of the loop start point. For looping in wavetable procedure, ADPCM sequencer must be reset.

ADPCM_LP_INDEX The 7-bit ADPCM index value of loop start point. Valid values ranged from 0 to 88.

Note_x+0024h Word 9 in the working memory WORK_MEM_9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**IIR_PRE_Y1[17:16]**	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IIR_PRE_Y1[15:0]**															
Type	R/W															

IIR_PRE_Y1 Previous one filtered value Y(n-1) used in IIR filter

Note_x+0028h Word 10 in the working memory WORK_MEM_10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name															IIR_PRE_Y2[17:16]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IIR_PRE_Y2[15:0]															
Type	R/W															

IIR_PRE_Y1 Previous two filtered value Y(n-2) used in IIR filter

Note_x+002Ch Word 11 in the working memory WORK_MEM_11

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IIR_A0															
Type	R/W															

IIR_A0 IIR filter parameter A0. The filter formula is Y(n) = A0*X(n) + 2B1*Y(n-1) – B2*Y(n-2).

Note_x+0030h Word 12 in the working memory WORK_MEM_12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IIR_B1															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IIR_B2															
Type	R/W															

IIR_B1 IIR filter parameter B1.

IIR_B2 IIR filter parameter B2.

Note_x+0034h Word 13 in the working memory WORK_MEM_13

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PAN_PAR_L															
Type	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PAN_PAR_R															
Type	R/W															

PAN_PAR_R right channel pan calculation parameter. It is a fractional number in unsigned 0.16 format.

PAN_PAR_L left channel pan calculation parameter. It is a fractional number in unsigned 0.16 format.

6 Clock, Mixed Subsystem

6.1 Analog Front-end & Analog Blocks

6.1.1 General Description

To communicate with analog blocks, a common control interface for all analog blocks is implemented. In addition, there are some dedicated interfaces for data transfer. The common control interface translates APB bus write and read cycle for specific addresses related to analog front-end control. During writing or reading of any of these control registers, there is a latency associated with transferring of data to or from the analog front-end. Dedicated data interface of each analog block is implemented in the corresponding digital block. The Analog Blocks includes the following analog function for complete GSM/GPRS base-band signal processing:

1. *Base-band RX*: For I/Q channels base-band A/D conversion
2. *Base-band TX*: For I/Q channels base-band D/A conversion and smoothing filtering, DC level shifting
3. *RF Control*: Two DACs for automatic power control (APC) and automatic frequency control (AFC) are included. Their outputs are provided to external RF power amplifier and VCXO), respectively.
4. *Auxiliary ADC*: Providing an ADC for battery and other auxiliary analog function monitoring
5. *Audio mixed-signal blocks:* It provides complete analog voice signal processing including microphone amplification, A/D conversion, D/A conversion, earphone driver, and etc. Besides, dedicated stereo D/A conversion and amplification for audio signals are included).
6. *Clock Generation*: A clock squarer for shaping system clock, and six PLLs that provide clock signals to DSP, MCU, Camera, TVOUT, Memory card, CEVA DSP and USB units are included
7. *XOSC32*: It is a 32-KHz crystal oscillator circuit for RTC application Analog Block Descriptions

6.1.1.1 BBRX

6.1.1.1.1 Block Descriptions

The receiver (RX) performs base-band I/Q channels downlink analog-to-digital conversion:

1. *Analog input multiplexer:* For each channel, a 4-input multiplexer that supports offset and gain calibration is included.
2. *A/D converter:* Two 14-bit sigma-delta ADCs perform I/Q digitization for further digital signal processing.

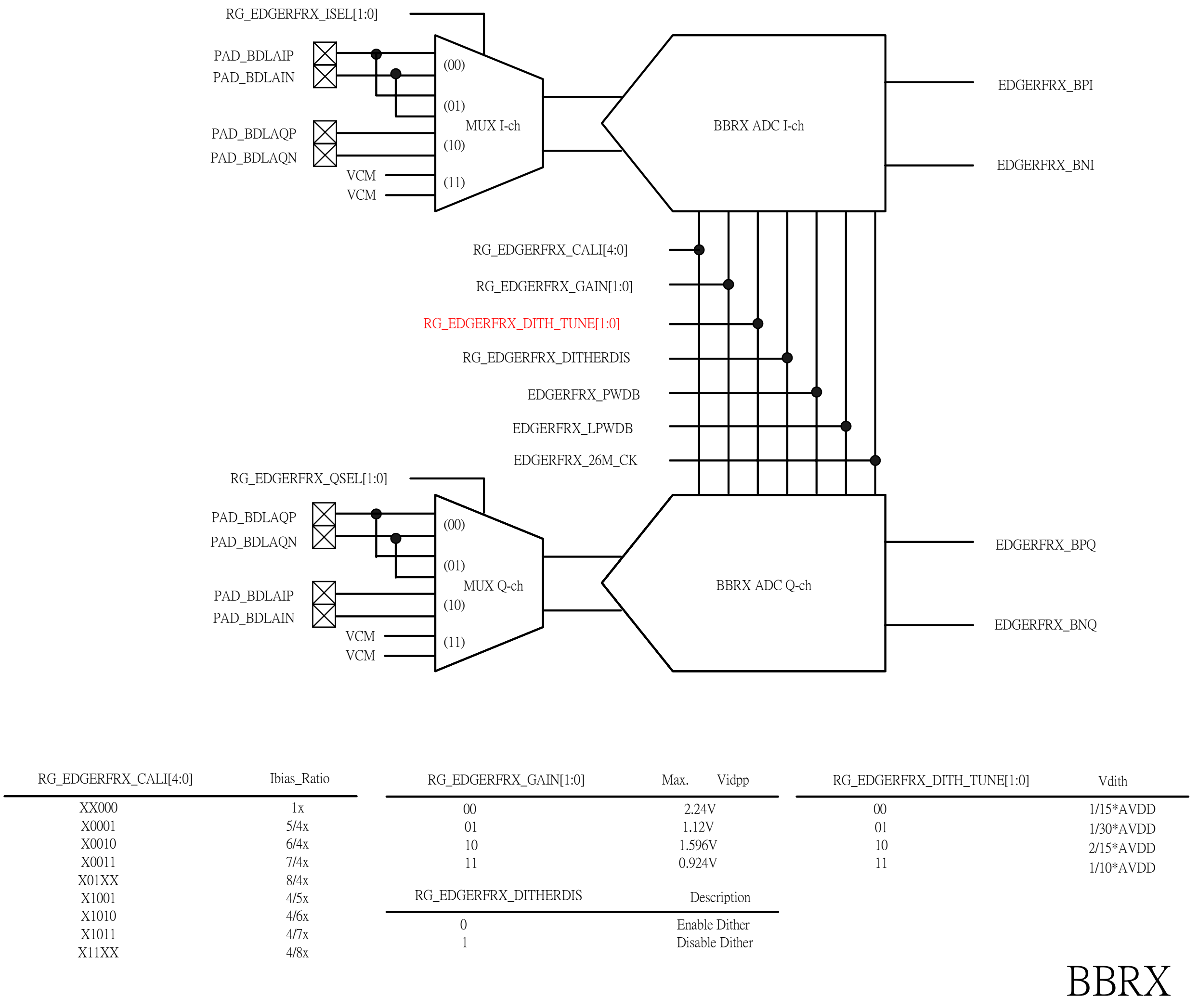


Figure 149 Block diagram of BBRX

6.1.1.1.2 Functional Specifications

The functional specifications of the base-band downlink receiver are listed in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
N	Resolution		14		Bit
	Code Type		2's complemen t		
FC	Clock Rate		26		MHz
FS	Output Sampling Rate		13/12		MSPS
	Input Swing		0.8*AVDD		Vpk

	When GAIN='00' When GAIN='01' When GAIN='10' When GAIN='11'		0.4*AVDD 0.57*AVDD 0.33*AVDD		
OE	Offset Error		+/- 10		mV
FSE	Full Swing Error		+/- 30		mV
	I/Q Gain Mismatch			0.5	dB
SINAD	Signal to Noise and Distortion Ratio - 45kHz sine wave in [0:90] kHz bandwidth - 145kHz sine wave in [10:190] kHz bandwidth	70 70			dB dB
ICN	Idle channel noise - [0:90] kHz bandwidth - [10:190] kHz bandwidth			-74 -70	dB dB
DR	Dynamic Range - [0:90] kHz bandwidth - [10:190] kHz bandwidth	74 70			dB dB
RIN	Input Resistance	75			kΩ
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption Power-up Power-Down		1 1		mA μA

*Table **125 Base-band Downlink Specifications***

6.1.1.2 BBTX

6.1.1.2.1 Block Descriptions

The transmitter (TX) performs base-band I/Q channels up-link digital-to-analog conversion. Each channel includes:

1. *10-Bits D/A Converter:* It converts digital modulated signals to analog domain. The input to the DAC is sampled at 4.33-MHz rate with 10-bits resolution.
2. *Smoothing Filter:* The low-pass filter performs smoothing function for DAC output signals with a 350-kHz 3rd-order Butterworth frequency response for GSM/GPRS..

6.1.1.2.2 Function Specifications

The functional specifications of the base-band uplink transmitter are listed in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
N	Resolution		10		Bit

	Code type		2's complement		
FS	Sampling Rate GSM/GPRS Mode		4.33		MSPS MSPS
SINAD	Signal to Noise and Distortion Ratio	57	60		dB
	Output Swing	0.21*AVDD		0.52*AVDD	V
VOCM	Output CM Voltage	0.3*AVDD	0.5*AVDD	0.7*AVDD	V
	Output Capacitance			20	PF
	Output Resistance	10			KΩ
DNL	Differential Nonlinearity	-0.5		+0.5	LSB
INL	Integral Nonlinearity	-0.5		+0.5	LSB
OE	Offset Error		+/- 15		mV
FSE	Full Swing Error		+/- 30		mV
FCUT	Filter –3dB Cutoff Frequency GSM/GPRS Mode	300	350	400	KHz
ATT	Filter Attenuation (GSM/GPRS Mode) at 100-KHz 270-KHz 4.33-MHz		0.0 0.84 65.72		dB dB dB
	I/Q Gain Mismatch		+/- 0.5		dB
	I/Q Gain Mismatch Correction Range	-0.96		+0.84	dB
DVDD	Digital Power Supply	1.08	1.2	1.32	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption Power-up Power-Down		3 2		mA μA

*Table **126 Base-band Uplink Transmitter Specifications***

6.1.1.3 AFC-DAC

6.1.1.3.1 Block Descriptions

As shown in the following figure, AFC-DAC is designed to produce a single-ended output signal at AFC pin. AFC pin should be connected to an external 1st-order R-C low pass filter to meet the 13-bits resolution (DNL) requirement[1].

[1] DNL performance depends on external output RC filter bandwidth: the narrower the bandwidth, the better the DNL. Thus, there exists a tradeoff between output setting speed and DNL performance

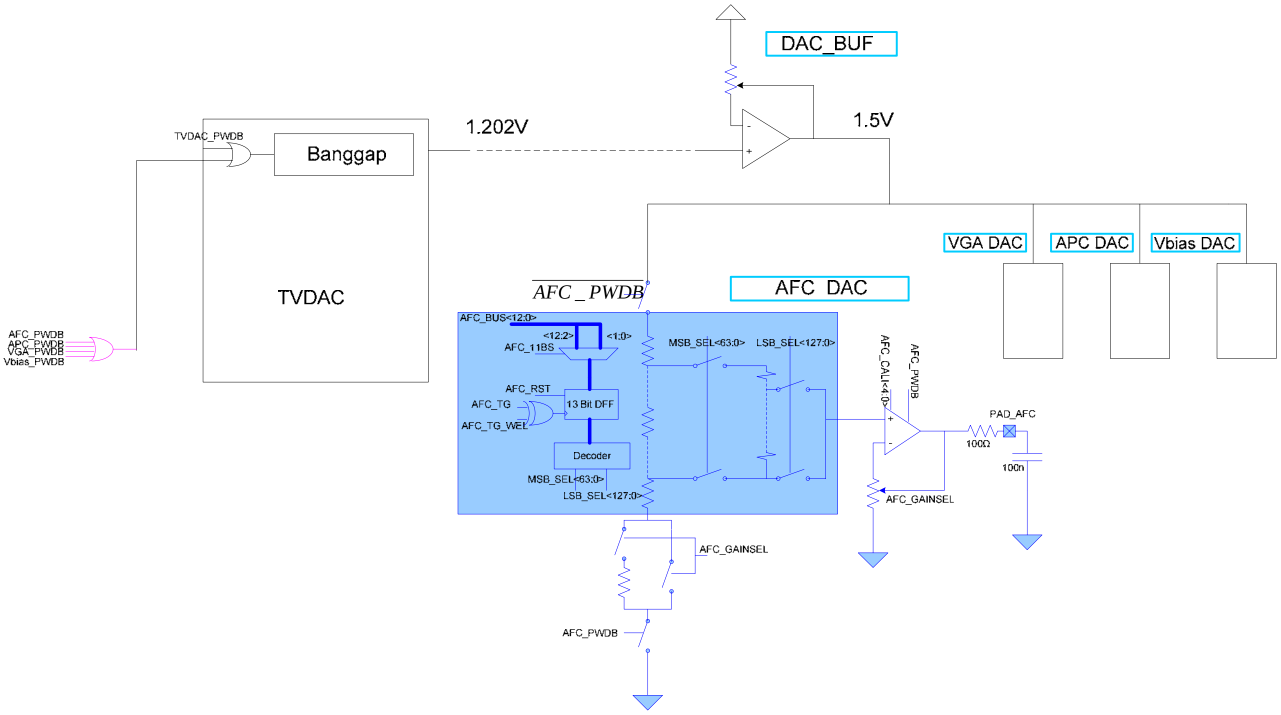


Figure 2 Block diagram of AFC-DAC

6.1.1.3.2 Functional Specifications

The following table gives the electrical specification of AFC-DAC.

Symbol	Parameter	Min	Typical	Max	Unit
N	Resolution		13		Bit
	Code type		Offset binary		
FS	Sampling Rate		1083		KHz
DVDD	Digital Power Supply	1	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20	60	125	℃
	Current Consumption Power-up Power-Down			500 1	μA μA
	Output Range Gainsel=0 Gainsel=1		AVDD/2±3/8 *AVDD 0~AVDD		V
	Output Resistor (in AFC output RC network)		100		Ω
DNL	Differential Nonlinearity		+1/-1		LSB
INL	Integral Nonlinearity		+5/-5		LSB

© 2009 MediaTek Inc.

	Settling time		166		μs
	Maximum code difference		70		LSB
PSR	Power supply ripple rejection ratio		-57		dB

Table 127 ***Functional specification of AFC-DAC***

6.1.1.4 APC-DAC

6.1.1.4.1 Block Descriptions

The APC-DAC is a 10-bits DAC with output buffer aimed for automatic power control. Here blow are its analog pin assignment and functional specification tables.

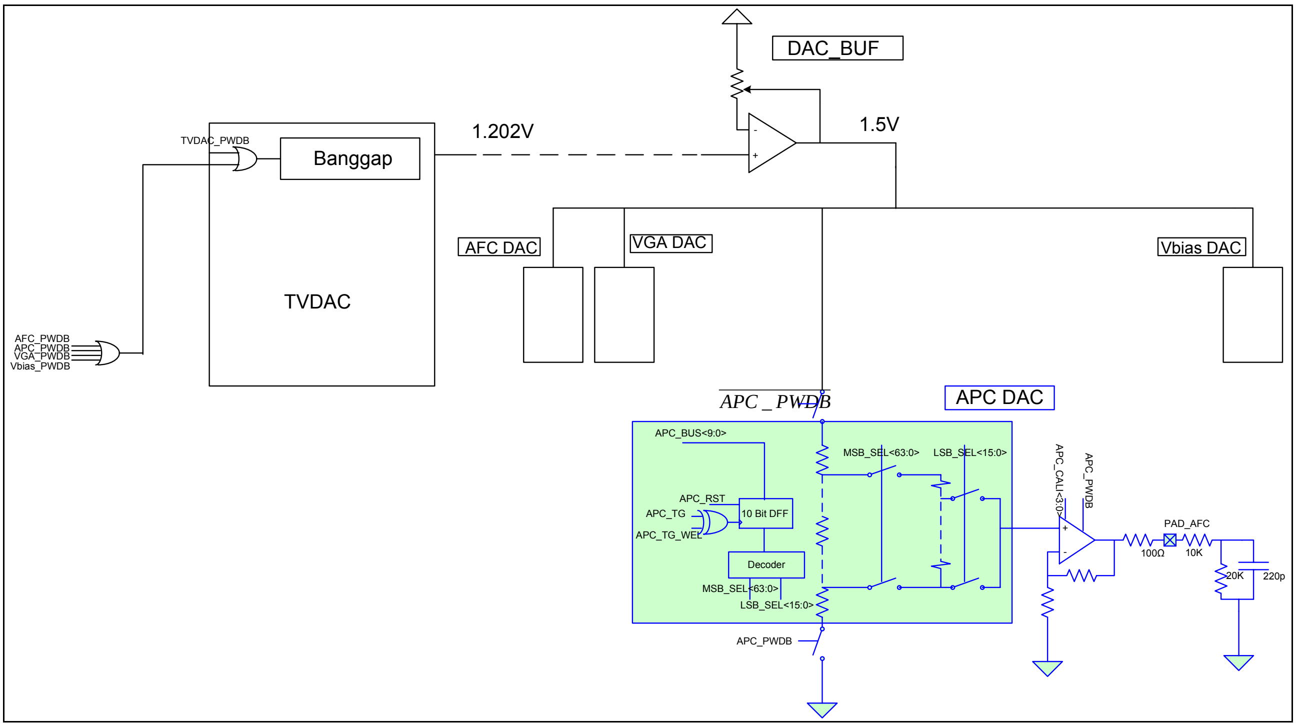


Figure 3 Block diagram of APC-DAC

6.1.1.4.2 Function Specifications

Symbol	Parameter	Min	Typical	Max	Unit
N	Resolution		10		Bit
	Code Type		Offset binary		
FS	Sampling Rate			1.0833	MSPS
SINAD	Signal to Noise and Distortion Ratio (10-KHz Sine with 1.0V Swing & 100-KHz BW)		50		dB
	99% Settling Time (Full Swing on Maximal Capacitance)			5	μS

	Output Swing			AVDD-0.2	V
	Output Capacitance			200	pF
	Output Resistance	10			KΩ
DNL	Differential Nonlinearity		+/- 0.5		LSB
INL	Integral Nonlinearity		+/- 1.0		LSB
OE	Offset Error		+/- 10		mV
FSE	Full Swing Error		+/- 10		mV
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20	60	125	℃
	Current Consumption Power-up Power-Down		200 1		μA μA

Table 128 APC-DAC Specifications

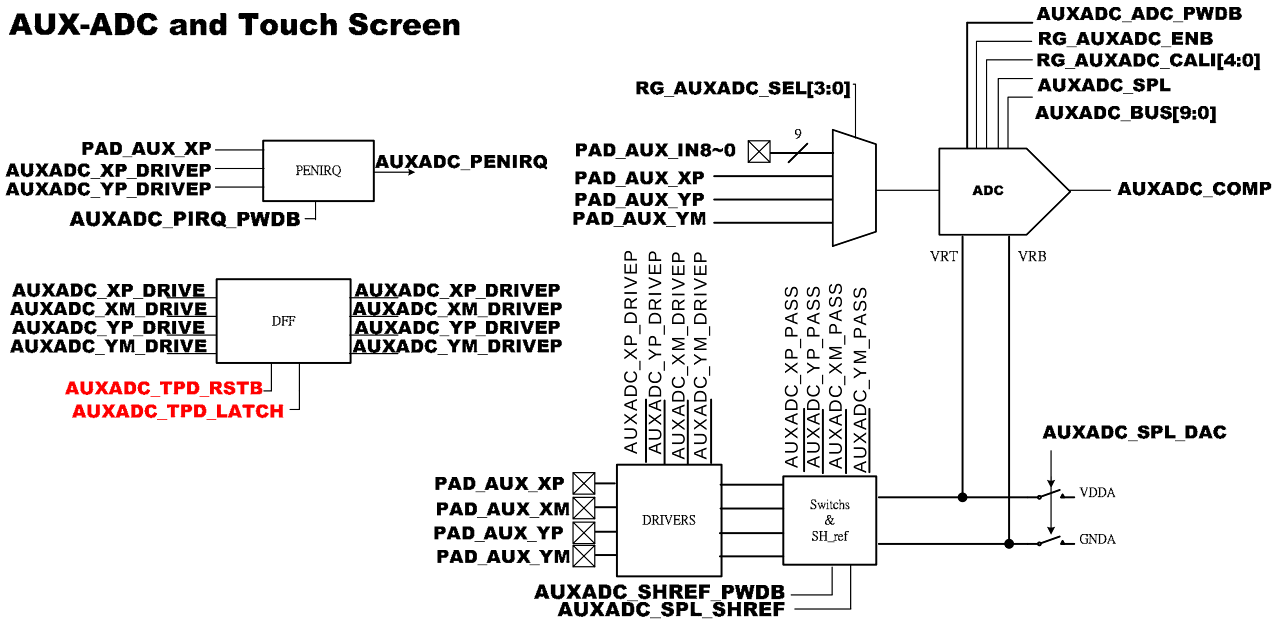
6.1.1.5 Auxiliary ADC

6.1.1.5.1 Block Descriptions

The auxiliary ADC includes the following functional blocks:

1. *Analog Multiplexer:* The analog multiplexer selects signal from one of the seven auxiliary input pins. Real word message to be monitored, like temperature, should be transferred to the voltage domain.
2. *10 bits A/D Converter:* The ADC converts the multiplexed input signal to 10-bit digital data.

© 2009 MediaTek Inc.



RG_AUXADC_ENB	Offset cancellation function
0	enable
1	disable

RG_AUXADC_CALI[4:0]	Ibias ratio
XX000	1x
0X001	5/4x
0X010	6/4x
0X011	7/4x
0X1XX	8/4x
1X001	4/5x
1X010	4/6x
1X011	4/7x
1X1XX	4/8x

Figure 4 Block diagram of AUX ADC

6.1.1.5.2 Function Specifications

The functional specifications of the auxiliary ADC are listed in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
N	Resolution		10		Bit
	Code Type		Offset binary		
FC	Clock Rate	0.1		2.56	MHz
FS	Sampling Rate @ N-Bit			2.56/(N+1)	MSPS
	Input Swing	0.0		AVDD	V
CIN	Input Capacitance Unselected Channel Selected Channel			 50 1.2	fF pF
RIN	Input Resistance				

	Unselected Channel Selected Channel	10 1.8			MΩ MΩ
	Clock Latency		11		1/FC
DNL	Differential Nonlinearity		+1.0/-1.0		LSB
INL	Integral Nonlinearity		+1.0/-1.0		LSB
OE	Offset Error		+/- 10		mV
FSE	Full Swing Error		+/- 10		mV
SINAD	Signal to Noise and Distortion Ratio (10-KHz Full Swing Input & 13-MHz Clock Rate)	50			dB
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption Power-up Power-Down		300 1		μA μA

Table 129 ***The Functional specification of Auxiliary ADC***

6.1.1.6 Audio mixed-signal blocks

6.1.1.6.1 Block Descriptions

Audio mixed-signal blocks (AMB) integrate complete voice uplink/downlink and audio playback functions. As shown in the following figure, it includes mainly three parts. The first consists of stereo audio DACs and speaker amplifiers for audio playback. The second is the voice downlink path, including voice-band DAC and amplifier, which produces voice signal. Amplifiers in audio blocks are equipped with multiplexers to accept signals from internal audio/voice or external radio sources. The last is the voice uplink path, which is the interface between microphone (or other auxiliary input device) input and MT6516 DSP. A set of bias voltage is provided for external electrical microphone.

 © 2009 MediaTek Inc.

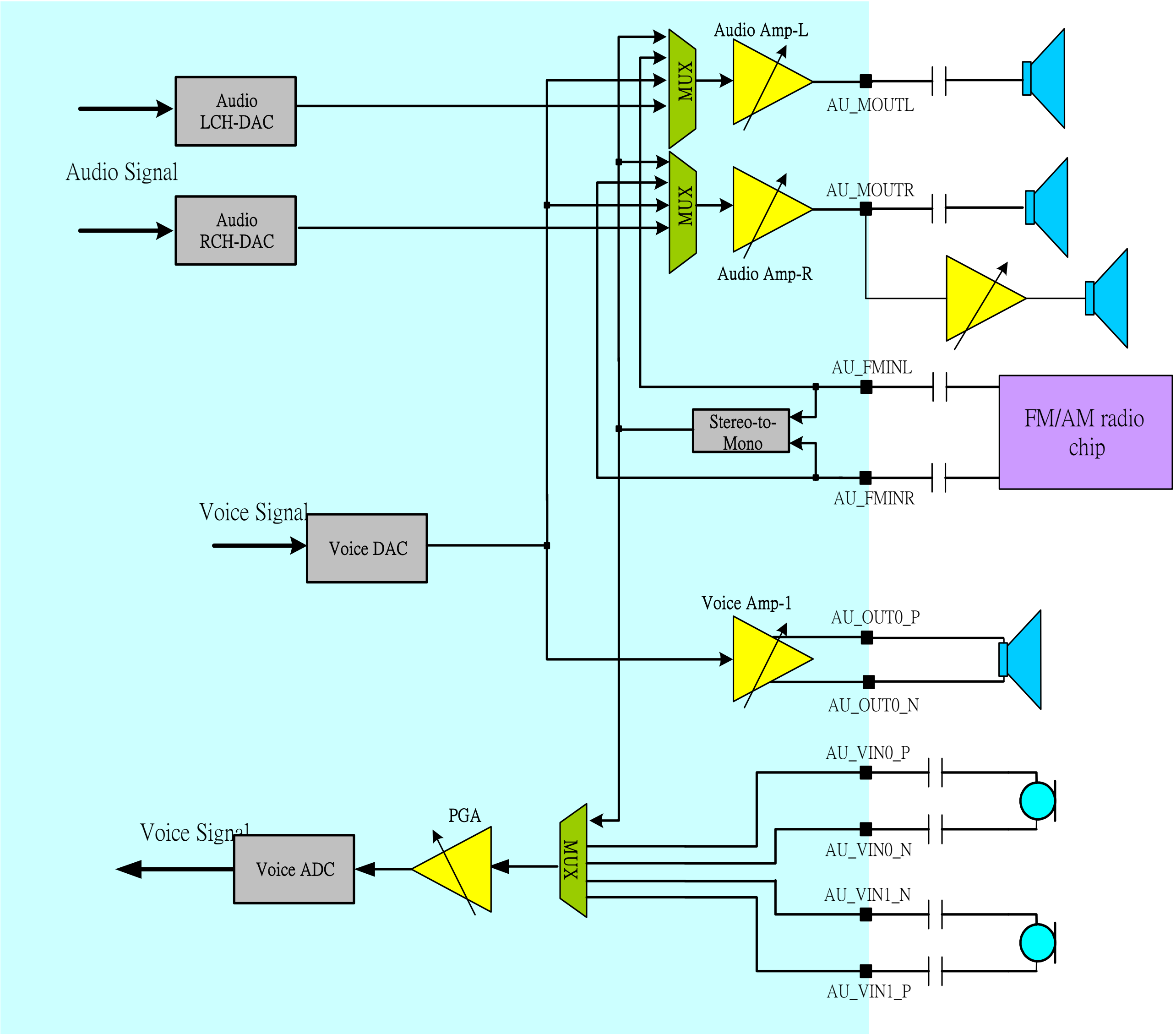


Figure 5 Block diagram of audio mixed-signal blocks.

6.1.1.6.2 Functional Specifications

The following table gives functional specifications of voice-band uplink/downlink blocks.

Symbol	Parameter	Min	Typical	Max	Unit
FS	Sampling Rate		6500		KHz
CREF	Decoupling Cap Between AU_VCM_PO And AU_VCM_NO		1		uF
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V

T	Operating Temperature	-20		80	℃
IDC	Current Consumption of uplink		2		mA
	Current Consumption of downlink		2.85		mA
VMIC	Microphone Biasing Voltage		1.9	2.2	V
IMIC	Current Draw From Microphone Bias Pins			2	mA
Uplink Path[2]					
SINAD	Signal to Noise and Distortion Ratio Input Level: -40 dbm0 Input Level: 0 dbm0 (200mVrms differential)	29	69		dB dB
RIN	Input Impedance (Differential)	13	20	27	KΩ
ICN	Idle Channel Noise			-67	dBm0
XT	Crosstalk Level			-66	dBm0
Downlink Path[3]					
SINAD	Signal to Noise and Distortion Ratio Input Level: -40 dBm0 Input Level: 0 dBm0	29	69		dB dB
RLOAD	Output Resistor Load (Differential)	28			Ω
CLOAD	Output Capacitor Load			200	pF
ICN	Idle Channel Noise of Transmit Path			-67	dBm0
XT	Crosstalk Level on Transmit Path			-66	dBm0

Table 130 ***Functional specifications of analog voice blocks***

Functional specifications of the audio blocks are described in the following.

Symbol	Parameter	Min	Typical	Max	Unit
FCK	Clock Frequency		6500		KHz
Fs	Sampling Rate	32	44.1	48	KHz
AVDD	Power Supply	2.6	2.8	3.1	V
T	Operating Temperature	-20		80	℃
IDC	Current Consumption		5		mA

[2] For uplink-path, not all gain setting of VUPG meets the specification listed on table, especially for the several lowest gains. The minimum gain that meets the specification is to be determined.

[3] For uplink-path, not all gain setting of VUPG meets the specification listed on table, especially for the several lowesthighest gains. The maximum minimum gain that meets the specification is to be determined.
[3] For downlink-path, not all gain setting of VDPG meets the specification listed on table, especially for the several lowest gains. The minimum gain that meets the specification is to be determined.

PSNR	Peak Signal to Noise Ratio		85		dB
IDC	**Current Consumption**		6		mA
DR	Dynamic Range		90		dB
VOUT	Output Swing for 0dBFS Input Level		0.85		Vrms
THD	Total Harmonic Distortion 45mW at 16 Ω Load 22mW at 32 Ω Load			-85 -85	 dB dB
RLOAD	Output Resistor Load (Single-Ended)	16			Ω
CLOAD	Output Capacitor Load			200	pF
XT	L-R Channel Cross Talk			TBD	dB

Table 131 ***Functional specifications of the analog audio blocks***

6.1.1.7 Clock Squarer

6.1.1.7.1 Block Descriptions

For most VCXO, the output clock waveform is sinusoidal with too small amplitude (about several hundred mV) to make MT6516 digital circuits function well. Clock squarer is designed to convert such a small signal to a rail-to-rail clock signal with excellent duty-cycle. It provides also a pull-down function when the circuit is powered-down.

6.1.1.7.2 Function Specifications

The functional specification of clock squarer is shown in Table 132.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency	13	26		MHz
Fout	Output Clock Frequency	13	26		MHz
Vin	Input Signal Amplitude		500	AVDD	mVpp
DcycIN	Input Signal Duty Cycle		50		%
DcycOUT	Output Signal Duty Cycle	DcycIN-5		DcycIN+5	%
TR	Rise Time on Pin CLKSQOUT			5	ns/pF
TF	Fall Time on Pin CLKSQOUT			5	ns/pF
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		100		uA

Table 132 The Functional Specification of Clock Squarer

6.1.1.7.3 Application Notes

Here below in the figure is an equivalent circuit of the clock squarer. Please be noted that the clock squarer is designed to accept a sinusoidal input signal. If the input signal

© 2009 MediaTek Inc.

is not sinusoidal, its harmonic distortion should be low enough to not produce a wrong clock output. As a reference, for a 13MHz sinusoidal signal input with amplitude of 0.2V the harmonic distortion should be smaller than 0.02V.

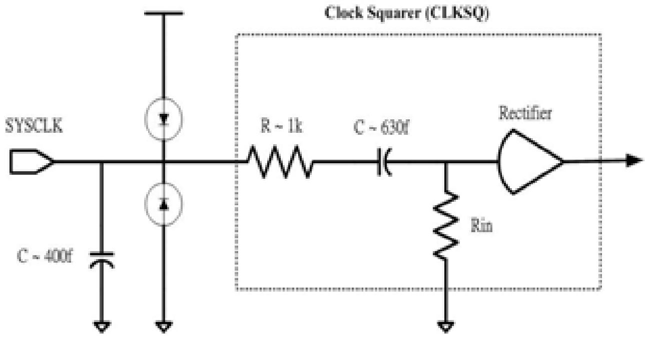


Figure 6 Equivalent circuit of Clock Squarer.

6.1.1.8 Phase Locked Loop

6.1.1.8.1 Block Descriptions

MT6516 includes six PLLs: MCU PLL, USB PLL, CEVA DSP PLL, Camera PLL, Memory card PLL and TV PLL. USB PLL is designed to accept 13MHz input clock signal and provides 48MHz output clocks for USB and IRDA. MCU PLL is designed to accept 13MHz input clock signal and provides both 416MHz output clock for MCU domain and 104MHz clock for DSP domain. At the same time, MCPLL and TV PLL are designed to accept 13MHz input clock signal and provides 91MHz and 27MHz output clock, respectively. Camera PLL are programmable to provide 104~208MHz output clock. Likewise, CEVA PLL are programmable to provide 208~416MHz output clock.

© 2009 MediaTek Inc.

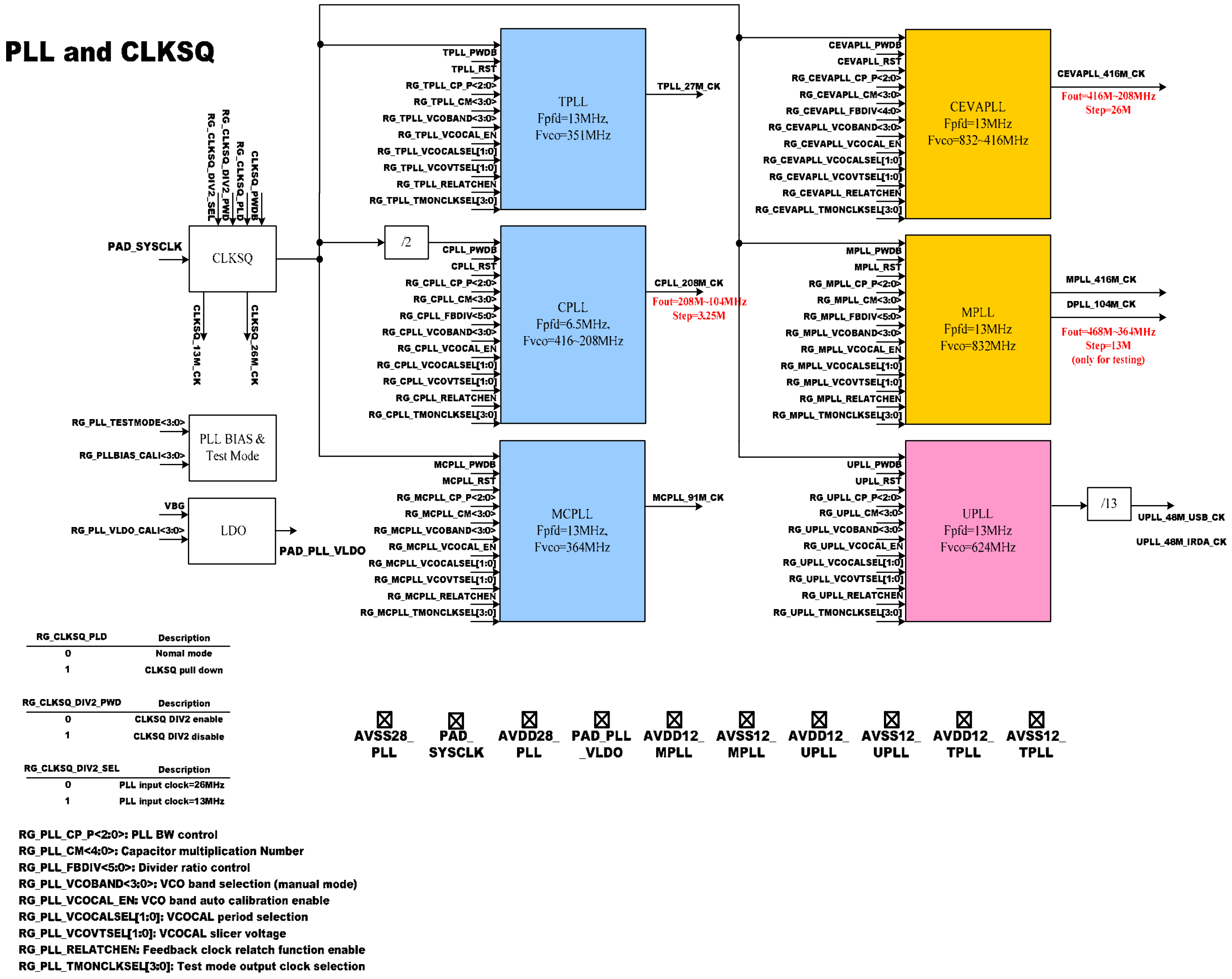


Figure 7 ***PLL & CLKSQ Block Diagram***

6.1.1.8.2 Function Specifications

The functional specification of MCU PLL is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		13		MHz
Fout	Output Clock Frequency	MCU: DSP:	416 104		MHz
	Lock-in Time		200		us

	Output Clock Duty Cycle	45	50	55	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		mA

Table 133 The Functional Specification of MCU PLL

The functional specification of USB PLL is shown below.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		13		MHz
Fout 1	Output Clock Frequency (for USB &IRDA)		48		MHz
	Lock-in Time		200		μs
	Output Clock Duty Cycle	40	45	50	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		μA

Table 134 The Functional Specification of USB PLL

The functional specification of Camera PLL is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		6.5		MHz
Fout	Output Clock Frequency	104		208	MHz
	Lock-in Time		400		us
	Output Clock Duty Cycle	45	50	55	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		mA

Table 135 The Functional Specification of Camera PLL

The functional specification of TV PLL is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		13		MHz

Fout	Output Clock Frequency		27		MHz
	Lock-in Time		200		us
	Output Clock Duty Cycle	40	45	50	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.6	2.8	3.0	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		mA

Table 136 The Functional Specification of TV PLL

The functional specification of MC PLL is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		13		MHz
Fout	Output Clock Frequency		91		MHz
	Lock-in Time		200		us
	Output Clock Duty Cycle	45	50	55	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		mA

Table 137 The Functional Specification of MC PLL

The functional specification of CEVA PLL is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
Fin	Input Clock Frequency		13		MHz
Fout	Output Clock Frequency	208	312	416	MHz
	Lock-in Time		200		us
	Output Clock Duty Cycle	45	50	55	%
	Output Clock Jitter	-100		+100	ps
DVDD	Digital Power Supply	1.0	1.2	1.4	V
AVDD	Analog Power Supply	2.5	2.8	3.1	V
T	Operating Temperature	-20		80	℃
	Current Consumption		TBD		mA

Table 138 The Functional Specification of CEVA PLL

© 2009 MediaTek Inc.

6.1.1.9 32-KHz Crystal Oscillator

6.1.1.9.1 Block Descriptions

The low-power 32-KHz crystal oscillator XOSC32 is designed to work with an external piezoelectric 32.768kHz crystal and a load composed of two functional capacitors, as shown in the following figure.

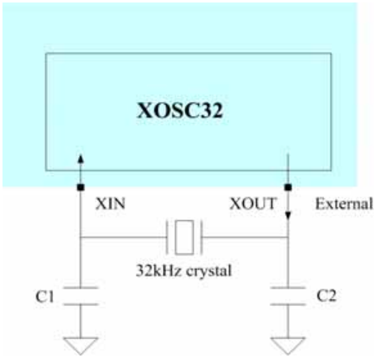


Figure 8 Block diagram of XOSC32

6.1.1.9.2 Functional specifications

The functional specification of XOSC32 is shown in the following table.

Symbol	Parameter	Min	Typical	Max	Unit
AVDDRTC	Analog power supply	0.9	2.8	3.0	V
Tosc	Start-up time			5	sec
Dcyc	Duty cycle	35	50		%
TR	Rise time on XOSCOUT		TBD		ns/pF
TF	Fall time on XOSCOUT		TBD		ns/pF
	Current consumption			5	µA
	Leakage current		1		µA
T	Operating temperature	-20		80	℃
gm	Transconductance of OSC32	15			µA/V

Table 139 Functional Specification of XOSC32

Here below are a few recommendations for the crystal parameters for use with XOSC32.

Symbol	Parameter	Min	Typical	Max	Unit
F	Frequency range		32768		Hz
GL	Drive level			5	uW
Δf/f	Frequency tolerance			+/- 20	Ppm
ESR	Series resistance			50	KΩ
C0	Static capacitance			1.6	pF

 © 2009 MediaTek Inc.

CL[4]	Load capacitance			27	pF

Table 140 Recommended Parameters of the 32kHz crystal

6.1.1.10 SIM Card Interface

6.1.1.10.1 Block Descriptions

The SIM card interface circuitry of MT6516 meets all ETSI and IMT-2000 SIM interface requirements. It provides level shifting needs for low voltage GSM controller to communicate with either 1.8V or 3V SIM cards. All SIM cards contain a clock input, a reset input, and a bi-directional data input/output. The clock and reset inputs to SIM cards are level shifted from the supply of digital IO (Vio) of baseband chipset to the SIM supply (Vsim). The bi-directional data bus is internal pull high with 10kohm resistor.

All pins that connect to the SIM card (Vsim, SRST, SCLK, SIO) withstand over ??(5kV) of human body mode ESD. In order to ensure proper ESD protection, careful board layout is required.

6.1.1.10.2 Functional specifications

Symbol	Parameter	Min	Typical	Max	Unit
Interface to 3 V SIM Card					
Volrst	The lowest output level of the SRST (I = 20 μA)	---	---	0.4	V
Vohrst	The highest output level of the SRST (I = -200 μA)	0.9*VSIM	---	---	V
Volclk	The lowest output level of the SCLK (I = 20 μA)	---	---	0.4	V
Vohclk	The highest output level of the SCLK (I = -200 μA)	0.9*VSIM	---	---	V
Vilsio	The maximum input low level which the SIO can accept r	---	---	0.4	V
Vihsio , Vohsio	The minimum input/output high level which SIO can accept/ apply (I = ±20 μA)	VSIM-0.4	---	---	V
Iilsio	The maximum current drawn out from the SIO while the SIO input voltage is low (Vilsio = 0 V)	---	---	-1	mA
Volsio	The maximum output low level which SIO can apply (Iolsio = 1 mA, SIMIO ≤ 0.23 V)	---	---	0.15*VSIM	V
Interface to 1.8 V SIM Card					
Volrst	SRST the lowest output level (I = 20 μA)			0.2*VSIM	V
Vohrst	SRST the highest output level	0.9*VSIM			V

[4] CL is the parallel combination of C1 and C2 in the block diagram.

 © 2009 MediaTek Inc.

	(I = -200 μA)				
Volclk	SCLK the lowest output level (I = 20 μA)			0.2*VSIM	V
Vohclk	SCLK the highest output level (I = -200 μA)	0.9*VSIM			V
Vilsio	The maximum input level which SIO can accept			0.15*VSIM	V
Vihsio , Vohsio	The minimum input/output high level which SIO can accept/ apply (I = ±20 μA)	VSIM-0.4			V
Iilsio	The maximum current drawn out from the SIO while the SIO input voltage is low (Vilsio = 0 V)			-1	mA
Volsio	The maximum output low level which SIO can apply (Iolsio = 1 mA, SIMIO ≤ 0.23 V)			0.15*VSIM	V
SIM Card Interface Timing					
	SIO pull-up resistance to VSIM	8	10	12	kΩ
	SRST, SIO rise/fall times (VSIM = 3, 1.8 V, load with 30 pF)			1	μs
	SCLK rise/fall times (VSIM = 3 V, CLK load with 30 pF)			18	ns
	(VSIM = 1.8 V, CLK load with 30 pF)			50	ns
	SCLK frequency (CLK load with 30 pF)	5			MHz
	SCLK duty cycle (SIMCLK Duty = 50%, fsimclk = 5 MHz)	47		53	%
	SCLK propagation delay		30	50	ns

Table 19 ***Functional Specification of SIM Card Level-shift***

6.1.2 MCU Register Definitions

Table20 list Control Registers and their address mapping to the MCU address space. The Audio-related Control register addresses starts from the base address 0x80060000. While other analog control register addresses starts from the base address 0x83010000.

Register Address	Register Function	Acronym
0x8301000C	Switch the Register Configuration Path	**CCI_WR_PATH**
0x83010100	Audio Front End Voice Analog Gain Control Register	**AFE_VAG_CON**
0x83010104	Audio Front End Voice Analog Circuit Control Register	**AFE_VAC_CON**
0x83010108	Audio Front End Voice Analog Circuit Control Register 1	**AFE_VAC_CON1**

 © 2009 MediaTek Inc.

0x8301010C	Audio Front End Voice Analog Power Down Control Register	**AFE_VAPDN_CON**
0x83010110	Audio Front End Voice Analog Circuit Control Register 2	**AFE_VAC_CON1**
0x80060200	Audio Front End Audio Analog Gain Control Register	**AFE_AAG_CON**
0x80060204	Analog Circuit Control Register	**AFE_AAC_CON**
0x80060208	Analog Circuit Control Register	**AFE_AAPDN_CON**
0x8006020C	Analog Circuit Control Register	**AFE_AAC_NEW**
0x80060210	Analog Circuit Control Register	**AFE_AAC_CON1**
0x83010300	BBRX ADC Analog Circuit Control Register	**BBRX_AC_CON**
0x83010310	WBRX ADC Analog Circuit Control Register	**WBRX_AC_CON**
0x83010400	BBTX DAC Analog Circuit Control Register 0	**BBTX_AC_CON0**
0x83010404	BBTX DAC Analog Circuit Control Register 1	**BBTX_AC_CON1**
0x83010408	BBTX DAC Analog Circuit Control Register 2	**BBTX_AC_CON2**
0x8301040C	BBTX DAC Analog Circuit Control Register 3	**BBTX_AC_CON3**
0x83010410	BBTX DAC Analog Circuit Control Register 4	**BBTX_AC_CON4**
0x83010500	AFC DAC Analog Circuit Control Register	**AFC_AC_CON**
0x83010600	APC DAC Analog Circuit Control Register	**APC_AC_CON**
0x83010700	AUX ADC Analog Circuit Control Register	**AUX_AC_CON**
0x83010900	TXVGA DAC Analog Circuit Control Register	**TXVGA_AC_CON**
0x83010A00	VBIAS DAC Analog Circuit Control Register	**VBIAS_AC_CON**

Table 20 ***Analog Control Registers***

6.1.2.1 BBRX

MCU APB bus registers for BBRX ADC are listed as followings.

0x83010300 BBRX ADC Analog-Circuit Control Register BBRX_AC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DITH_TUNE		DITHEN	QSEL		ISEL		RSV	GAIN		CALBIAS				
Type		RW		R/W	R/W		R/W		R/W	R/W		R/W				
Reset		00		0	00		00		0	00		00000				

Set this register for analog circuit configuration controls.

CALBIAS The register field is for control of biasing current in BBRX mixed-signal module. It is coded in 2's complement. That is, its maximum is 15 and minimum is –16. Biasing current in BBRX mixed-signal module has impact on the performance of A/D conversion. The larger the value of the register field, the larger the biasing current in BBRX mixed-signal module, and the larger the SNR.

GAIN The register bit is for configuration of gain control of analog inputs in GSM RX mixed-signal module.

- **00** Input range is 0.8x AVDD for analog inputs in GSM RX mixed-signal module.
- **01** Input range is 0.4x AVDD for analog inputs in GSM RX mixed-signal module.
- **10** Input range is 0.57x AVDD for analog inputs in GSM RX mixed-signal module.
- **11** Input range is 0.33x AVDD for analog inputs in GSM RX mixed-signal module.

ISEL Loopback configuration selection for I-channel in BBRX mixed-signal module

- **00** Normal mode
- **01** Loopback TX analog I
- **10** Loopback TX analog Q
- **11** Select the grounded input

QSEL Loopback configuration selection for Q-channel in BBRX mixed-signal module

- **00** Normal mode
- **01** Loopback TX analog Q
- **10** Loopback TX analog I
- **11** Select the grounded input

DITHDIS Dither feature Disable control register, which can effectively reduce the THD (total harmonic distortion) of the BBRX ADC.

- **0** turn on the dither (default value)
- **1** Disable the dither

DITH_TUNE Dither voltage selection.

- **00** dither voltage=1/15*AVDD
- **01** dither voltage=1/30*AVDD
- **10** dither voltage=2/15*AVDD
- **11** dither voltage=1/10*AVDD

6.1.2.2 BBTX

MCU APB bus registers for BBTX DAC are listed as followings.

0x83010400 BBTX DAC Analog-Circuit Control Register 0 BBTX_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CALRCDONE	STARTCALRC	GAIN			CALRCSEL			TRIMI				TRIMQ			
Type	R	R/W	R/W			R/W			R/W				R/W			
Reset	0	0	000			000			0000				0000			

Set this register for analog circuit configuration controls. The procedure to perform calibration processing for smoothing filter in BBTX mixed-signal module is as follows:

1. Write 1 to the register bit STARTCALRC. Start calibration process.

 © 2009 MediaTek Inc.

2. Read the register bit CALRCDONE. If read as 1, then calibration process finished. Otherwise repeat the step.
3. Write 0 to the register bit STARTCALRC. Stop calibration process.
4. The result of calibration process can be read from the register field CALRCOUT of the register BBTX_AC_CON1. Software can set the value to the register field CALRCSEL for 3-dB cutoff frequency selection of smoothing filter in DAC of BBTX.

Remember to set the register field CALRCCONT of the register BBTX_AC_CON1 to 0xb before the calibration process. It only needs to be set once.

TRIMQ The register field is used to control gain trimming of Q-channel DAC in BBTX mixed-signal module. It is coded in 2's complement, that is, with maximum 7 and minimum –8.

TRIMI The register field is used to control gain trimming of I-channel DAC in BBTX mixed-signal module. It is coded in 2's complement, that is, with maximum 7 and minimum –8.

CALRCSEL The register field is for selection of cutoff frequency of smoothing filter in BBTX mixed-signal module. It is coded in 2's complement. That is, its maximum is 3 and minimum is –4.

GAIN The register field is used to control gain of DAC in BBTX mixed-signal module. It has impact on both of I- and Q-channel DAC in BBTX mixed-signal module. It is coded in 2's complement, that is, with maximum 3 and minimum –4.

STARTCALRC Whenever 1 is writing to the bit, calibration process for smoothing filter in BBTX mixed-signal module will be triggered. Once the calibration process is completed, the register bit CARLDONE will be read as 1.

CALRCDONE The register bit indicates if calibration process for smoothing filter in BBTX mixed-signal module has finished. When calibration processing finishes, the register bit will be 1. When the register bit STARTCALRC is set to 0, the register bit becomes 0 again.

0x8301 0404 BBTX DAC Analog-Circuit Control Register 1 BBTX_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CALRCOUT			FLOAT	CALRCCNT					CALBIAS				CMV		
Type	R			R/W	R/W					R./W				R/W		
Reset	-			0	00000					0000				000		

Set this register for analog circuit configuration controls.

CMV The register field is used to control common voltage in BBTX mixed-signal module. It is coded in 2's complement, that is, with maximum 3 and minimum –4.

CALBIAS It can control bias current from 0.5x to 2x Bias current is default value* (4+CALI[2:0])/4 or 4/(4+CALI[2:0]) depending on CALI[3]=0 or1

CALRCCNT Parameter for calibration process of smoothing filter in BBTX mixed-signal module. Default value is '22'. Note that it is **NOT** coded in 2's complement. Therefore the range of its value is from 0 to 31. Remember to set it to 0x16 before BBTX calibration process if clock sent to BBTX is 26Mhz. Otherwise, set to 0xb if clock is 13Mhz. It only needs to be set once. In MT6516, only 26MHz clock is available

FLOAT The register field is used to have the outputs of DAC in BBTX mixed-signal module float or not.

CALRCOUT After calibration processing for smoothing filter in BBTX mixed-signal module, a set of 3-bit value is obtained. It is coded in 2's complement.

0x83010408 BBTX DAC Analog-Circuit Control Register 2 BBTX_AC_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		BYPASS		DCCOARSEQ		DCCOARSEI		DAC_PTR				DWAEN			CALRCAUTO	CALRCOPEN
Type		R/W		R/W		R/W		R/W				R/W			R/W	R/W
Reset		00		00		00		0000				0			0	0

Set this register for analog circuit configuration controls.

CALRCOPEN The register field is used to control normal Mode(close loop) or debug mode (open loop) for BBTX comparator in mixed signal

0 normal Mode (close loop)

1 debug Mode (open Loop)

CALRCAUTO The register field is used to control the result of calibration process of smoothing filter can automatically load to control the smoothing filter or not.

0 Not auto load, need manual load (default)

1 Auto load

DWAEN The register field is used to turn on the DWA scheme of the BBTX DAC,

0 DWA scheme off (default)

1 DWA scheme on

DACPTR The register field is used to configured the staring pointer of 1 hot pulling of DINI//Q[15:0] signal to BBTX DAC, range from 0~15. There are two different configurations. For DWAEN = 0, pointer always starts from the configuration value (e.g. if DACPTR = 4'b1, 1 hot will start pulling from DINI/Q[1]). However, for DWAEN=1, the initial starting pointer will follow the configuration, while the pointer will move to most significant 1 hot pointer + 1 from the last DINI/Q[15:0] input. Defulat value is 0h.

DCCOARSEI The register field is used to control the central nominal value of BBTX DAC for I channel offset

00 central nominal @ +0LSB
01 central nominal @ +30LSB
11 central nominal @ - 30LSB
10 reserved

DCCOARSEQ The register field is used to control the central nominal value of BBTX DAC for Q channel offset

00 central nominal @ +0LSB
01 central nominal @ +30LSB
11 central nominal @ - 30LSB
10 reserved

BYPASS The register field is used to control the switch of bypass filter 1/2.

00 Default
01 DAC output to Pad

11 Not be used
10 Filter 1 output to Pad

0x8301040C BBTX DAC Analog-Circuit Control Register 3 BBTX_AC_CON3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MODE SEL	EN_2G	WB_GAIN			WB_CMV			WB_TRIMI				WB_TRIMQ			
Type	R/W	R/W	R/W			R/W			R/W				R/W			
Reset	0	0	000			000			0000				0000			

Set this register for analog circuit configuration controls for WCDMA (Since MT6516 does not support WCDMA mode, WB_GAIN, WB_CMV, WB_TRIMI,WB_TRIQ, WB_DCCOARSEI, and WB_DCCOARSEQ may be omitted.):

WB_TRIMQ The register field is used to control gain trimming of Q-channel DAC in BBTX mixed-signal module in WCDMA mode. It is coded in 2’s complement, that is, with maximum 7 and minimum –8.
For the analog circuit of BBTX, WB_TRIMQ is the valid configuration in WCDMA mode, while TRIMQ is the valid configuration in GSM/GPRS mode, respectively.

WB_TRIMI The register field is used to control gain trimming of I-channel DAC in BBTX mixed-signal module in WCDMA mode. It is coded in 2’s complement, that is, with maximum 7 and minimum –8.
For the analog circuit of BBTX, WB_TRIMI is the valid configuration in WCDMA mode, while TRIMI is the valid configuration in GSM/GPRS mode, respectively.

WB_GAIN The register field is used to control gain of DAC in BBTX mixed-signal module in WCDMA mode. It has impact on both of I- and Q-channel DAC in BBTX mixed-signal module. It is coded in 2’s complement, that is, with maximum 3 and minimum –4.
For the analog circuit of BBTX, WB_GAIN is the valid configuration in WCDMA mode, while GAIN is the valid configuration in GSM/GPRS mode, respectively.

WB_CMV The register field is used to control common voltage in BBTX mixed-signal module in WCDMA mode. It is coded in 2’s complement, that is, with maximum 3 and minimum –4. For the analog circuit of BBTX, WB_CMV is the valid configuration in WCDMA mode, while CMV is the valid configuration in GSM/GPRS mode, respectively.

EN_2G Software-controlled 2G mode enable. Note that this register field is only valid when MODESEL=1.
When MODESEL=1,
0 BBTX is in WCDMA mode. (default)
1 BBTX is in GSM/GPRS mode.

When MODESEL=0,
0 BBTX mode (GSM/GPRS or WCDMA) is under hardware control.
1 BBTX mode (GSM/GPRS or WCDMA) is under hardware control.

MODESEL The register field is used to select whether the BBTX analog circuit is
0 BBTX mode (GSM/GPRS or WCDMA) is under hardware control.
1 BBTX mode (GSM/GPRS or WCDMA) is controlled by the register field, EN_2G.

0x83010410 BBTX DAC Analog-Circuit Control Register 4 BBTX_AC_CON4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name													WB_DCCOARSEQ	WB_DCCOARSEI
Type													R/W	R/W
Reset													00	00

WB_DCCOARSEI The register field is used to control the central nominal value of BBTX DAC for I channel offset in WCDMA mode. For the analog circuit of BBTX, WB_DCCOARSEI is the valid configuration in WCDMA mode, while DCCOARSEI is the valid configuration in GSM/GPRS mode, respectively.

- **00** central nominal @ +0 LSB
- **01** central nominal @ +30 LSB
- **11** central nominal @ - 30 LSB
- **10** reserved

WB_DCCOARSEQ The register field is used to control the central nominal value of BBTX DAC for Q channel offset in WCDMA mode. For the analog circuit of BBTX, WB_DCCOARSEQ is the valid configuration in WCDMA mode, while DCCOARSEQ is the valid configuration in GSM/GPRS mode, respectively.

- **00** central nominal @ +0 LSB
- **01** central nominal @ +30 LSB
- **11** central nominal @ - 30 LSB
- **10** reserved

6.1.2.3 AFC DAC

MCU APB bus registers for AFC DAC are listed as follows.

0x83010500 AFC DAC Analog-Circuit Control Register AFC_AC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GAINSEL		TGSEL	11BS	CALI				
Type								R/W		R/W	R/W	R/W				
Reset								0		0	0	0				

Set this register for analog circuit configuration controls. Please refer to analog functional specification for more details.

GAINSEL gain selection of output swing

- 0 3/4*VDD
- 1 Full VDD

CALI Biasing current control.

- 0 00100 → 2 ×
- 1 00011 → 7/4 ×
- 2 00010 → 6/4 ×
- 3 00001 → 5/4 ×
- 4 00000 → 1×
- 5 10001 → 4/5 ×
- 6 10010 → 4/6 ×

7 10011 → 4/7 ×

8 10100 → 4/8 ×

11BS Test purpose. Degrade the resolution of AFC from 13 bits to 11 bits.

0 13 bits.

1 11 bits

TGSEL The register field is used to select whether the AFC DAC analog circuit is

0 Sample the 13-bit code at rising edge.

1 Sample the 13-bit code at falling edge.

6.1.2.4 APC DAC

MCU APB bus registers for APC DAC are listed as followings.

0x83010600 APC DAC Analog-Circuit Control Register APC_AC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DACBUF_CALI				DACBUF_TRIM					TGSEL	BYP		CALI			
Type	R/W				R/W					R/W	R/W		R/W			
Reset	0000				1000					0	0		0000			

Set this register for analog circuit configuration controls. Please refer to analog functional specification for more details.

DACBUF_TRIM Trimming bits for DAC supply buffer, including AFC DAC, APC DAC, TXVGA DAC, and VBIAS DAC.

DACBUF_CALI DAC supply buffer calibration bits, including AFC DAC, APC DAC, TXVGA DAC, and VBIAS DAC.

BYP bypass output buffer

0 Test DAC + output buffer.

1 Bypass output buffer.

CALI biasing current control

0 0100 → 2 ×

1 0011 → 7/4 ×

2 0010 → 6/4 ×

3 0001 → 5/4 ×

4 0000 → 1×

5 1001 → 4/5 ×

6 1010 → 4/6 ×

7 1011 → 4/7 ×

8 1100 → 4/8 ×

TGSEL The register field is used to select whether the APC DAC analog circuit is

0 Sample the 10-bit code at rising edge.

1 Sample the 10-bit code at falling edge.

6.1.2.5 Auxiliary ADC

MCU APB bus registers for AUX ADC are listed as followings.

0x83010700 Auxiliary ADC Analog-Circuit Control Register AUX_AC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										ENB		CALI				
Type										R/W		R/W				
Reset										0		0				

Set this register for analog circuit configuration controls. Please refer to analog functional specification for more details.

CALI Biasing current control

0 0100 → 2 ×

1 0011 → 7/4 ×

2 0010 → 6/4 ×

3 0001 → 5/4 ×

4 0000 → 1×

5 1001 → 4/5 ×

6 1010 → 4/6 ×

7 1011 → 4/7 ×

8 1100 → 4/8 ×

ENB Offset cancellation disable control.

0 Comparator offset cancellation function enable.

1 Comparator offset cancellation function disable.

6.1.2.6 Voice Front-end

MCU APB bus registers for speech are listed as followings.

0x83010100 AFE Voice Analog Gain Control Register AFE_VAG_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			VUPG						VDPG0				VZERO_DTC_CMP_PWD	VZERO_DTC_EN	VZERO_DTC_CALI	
Type			R/W						R/W				R/W	R/W	R/W	
Reset			00000						0000				0	0	00	

Set this register for analog PGA gains. VUPG is set for microphone input volume control. And VDPG0 and VDPG1 are set for two output volume controls

VUPG Voice-band up-link PGA gain control bits. For VCFG[3] = 1, it is only valid for INPUT 1. Note that this field only takes effect in analog test modes for testing purpose. Under normal operation, voice-band up-link PGA gain is controlled by audio front end. Please see the section “Audio Front End” for details.

VCFG[2] = ‘0’ Gain mode table				VCFG[2] = ‘1’ FM mode table			
VUPG [5:0]	Gain	VUPG [5:0]	Gain	VUPG [5:0]	Gain	VUPG [5:0]	Gain
111111	49 dB	011111	17 dB	111111	43 dB	011111	11dB
111110	48 dB	011110	16 dB	111110	42 dB	011110	10dB
111101	47 dB	011101	15 dB	111101	41 dB	011101	9dB
111100	46 dB	011100	14 dB	111100	40 dB	011100	8dB
111011	45 dB	011011	13 dB	111011	39 dB	011011	7dB
111010	44 dB	011010	12dB	111010	38 dB	011010	6dB
111001	43 dB	011001	11dB	111001	37 dB	011001	5dB
111000	42 dB	011000	10dB	111000	36 dB	011000	4dB
110111	41 dB	010111	9dB	110111	35 dB	010111	3dB
110110	40 dB	010110	8dB	110110	34 dB	010110	2dB
110101	39 dB	010101	7dB	110101	33 dB	010101	1dB
110100	38 dB	010100	6dB	110100	32 dB	010100	0dB
110011	37 dB	010011	5dB	110011	31 dB	010011	-1dB
110010	36 dB	010010	4dB	110010	30 dB	010010	-2dB
110001	35 dB	010001	3dB	110001	29 dB	010001	-3dB
110000	34 dB	010000	2dB	110000	28 dB	010000	-4dB
101111	33 dB	001111	1dB	101111	27 dB	001111	-5dB
101110	32 dB	001110	0dB	101110	26dB	001110	-6dB
101101	31 dB	001101	-1dB	101101	25 dB	001101	-7dB
101100	30 dB	001100	-2dB	101100	24 dB	001100	-8dB
101011	29 dB	001011	-3dB	101011	23 dB	001011	-9dB
101010	28 dB	001010	-4dB	101010	22 dB	001010	-10dB
101001	27 dB	001001	-5dB	101001	21 dB	001001	-11dB
101000	26dB	001000	-6dB	101000	20 dB	001000	-12dB
100111	25 dB	000111	-7dB	100111	19 dB	000111	-13dB
100110	24 dB	000110	-8dB	100110	18 dB	000110	-14dB
100101	23 dB	000101	-9dB	100101	17 dB	000101	-15dB
100100	22 dB	000100	-10dB	100100	16 dB	000100	-16dB
100011	21 dB	000011	-11dB	100011	15 dB	000011	-17dB
100010	20 dB	000010	-12dB	100010	14 dB	000010	-18dB
100001	19 dB	000001	-13dB	100001	13 dB	000001	-19dB

© 2009 MediaTek Inc.

100000	18 dB	000000	-14dB	100000	12dB	000000	-20dB

For VCFG[3] = 1, it is only valid for INPUT 1.

VDPG0 voice-band down-link PGA0 gain control bits

VDPG0 [3:0]	Gain
1111	8dB
1110	6dB
1101	4dB
1100	2dB
1011	0dB
1010	-2dB
1001	-4dB
1000	-6dB
0111	-8dB
0110	-10dB
0101	-12dB
0100	-14dB
0011	-16dB
0010	-18dB
0001	-20dB
0000	-22dB

VZERO_DTC_CALI Zero-detect comparator hysteresis adjust

- **0** 13mV
- **1** 26mV
- **2** 40mV
- **3** 56mV

VZERO_DTC_EN Voice buffer zero-detection enable.

- **0** disable
- **1** enable

VZERO_DTC_CMP_PWD Reserved.

0x83010104 AFE Voice Analog-Circuit Control Register 0 AFE_VAC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VDC_COUPLE	VMIC_SHORT	VMIC_VREF		VCFG					VTX_TESTMODE	VDSEND0	VCALI				
Type	R/W	R/W	R/W		R/W					R/W	R/W	R/W				
Reset	0	0	00		00000					0	00	00000				

Set this register for analog circuit configuration controls.

VTX_TESTMODE Config PAD_VIN1_P/ PAD_VIN1_N.

0 PAD_VIN1_P/PAD_VIN1_N are used for inputs from microphone.

1 PAD_VIN1_P/PAD_VIN1_N are configured to be VBITX PGA's differential outputs for testing.

VDC_COUPLE Selectively choose DC couple microphone sense. **(This cannot work in MT6516.)**

0 Disable DC couple sense of microphone

1 Enable DC couple sense of microphone

VMIC_SHORT Selectively short AU_MICBIASP / AU_MICBIASN.

0 float MIC_BIASN and short it to MIC_BIASP when handsfree mode mic is plugged in

1 short MIC_BIASN to ground when handsfree mode mic is plugged in. In this mode, differential mic has current leakage and cause power loss.

VMIC_VREF Tuning MICBIASP DC voltage.

00 1.9V

01 2.0V

10 2.1V

11 2.2V

VCFG[4] microphone biasing control

0 differential biasing

1 single-ended biasing

VCFG[3] gain mode control. This control register is only valid to input 1. Others can be amplification mode only.

0 amplification

1 bypass VBITX PGA to test ADC only

VCFG[2] coupling control

0 AC

1 DC

VCFG[1:0] input select control

00 input 0

01 input 1

10 FM

11 reserved

VDSEND0 Reserved

VCALI biasing current control, in 2's complement format

0x83010108 AFE Voice Analog-Circuit Control Register 1 AFE_VAC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VUPOP_EN	VBIAS_EN		VOC_EN	VADCINMODE	VCMBUF_EN	VCM_RLADDER_EN	VIBOOT	VFLOAT	VRSDON	VGBOOT	VADC_DVREF_CAL	VADC_DENB		VBUF_BIAS	
Type	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	
Reset	0	0		0	0	0	0	0	0	0	0	0	0		0	

Set this register for analog circuit configuration controls. There are several loop back modes and test modes implemented for test purposes. Suggested value is 0280h.

VUPOP_EN de-pop noise enable

0 disable

1 enable

VBIAS_EN voice downlink path bias enable

0 disable.

1 enable.

VOC_EN voice downlink buffer over current protection

0 disable

1 enable

VCM_RLADDER_EN VCM resistor ladder enable.

0 disable

1 enable

VCMBUF_EN Reference buffer enable.

0 disable

1 enable

VIBOOT Reserved

VFLOAT voice-band output driver float

0 normal operating mode

1 float mode

VRSDON voice-band redundant signed digit function on

0 1-bit 2-level mode

1 2-bit 3-level mode

VADC_DVREF_CAL ADC Dither Reference Voltage Calibration

0 3/15 VDD reference is fed to dither path

1 2/15 VDD reference is fed to dither path

VADC_DENB ADC Dither Enable

0 ADC dither enable

1 ADC dither disable

VADCINMODE Voice-band ADC output mode.

0 normal operating mode

1 the ADC input from the DAC output

VBUF_BIAS Voice downlink buffer output stage bias current adjust

0 3/3 x

1 4/3 x

2 1/3 x

3 2/3 x

0x8301010C AFE Voice Analog Power Down Control Register AFE_VAPDN_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									VPD_DCT_EN	VPDVCM_EN	VPDN_BIAS	VPDN_LNA	VPDN_ADC			VPDN_OUT0
Type									R/W	R/W	R/W	R/W	R/W			R/W
Reset									0	0	0	0	0			0

Set this register to power up analog blocks. 0: power down, 1: power up.

VPDVCM_EN Rreserved

VPD_DCT_EN DAC DCT circuit power on.

- **0** OFF
- **1** ON

VPDN_BIAS Mic-bias block power on.

- **0** OFF
- **1** ON

VPDN_LNA low noise amplifier block

- **0** OFF
- **1** ON

VPDN_ADC ADC block

- **0** OFF
- **1** ON

VPDN_OUT0 DAC Buffer.

- **0** OFF
- **1** ON

0x83010110 AFE Voice Analog-Circuit Control Register 2 AFE_VAC_CON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLOOPBACK	VRXCALI					VCLK2X_INV_EN	VCLK_INV_EN	VCLK_SEL_EN				VBIAS_DCT_EN			
Type	R/W	R/W					R/W	R/W	R/W				R/W			
Reset	0	00000					0	0	0000				0000			

Set this register for analog circuit configuration controls.

VLOOPBACK Voice Tx and Rx loop back mode.

- **0** Disable loop back mode.
- **1** Enable loop back mode.

VBIAS_DCT_EN BIAS current tunning register for DCT.

0~15 corresponds to 8/8,7/8,6/8,5/8,4/8,3/8,2/8,1/8,16/8,15/8,14/8,13/8,12/8,11/8,10/8,9/8

VCLK_SEL_EN Non-overlap clock timing tunning register

- **[3:2]** CLOCK DELAY FINE TUNE
- **[1:0]** NON-OVERLAPPING CLOCK FINE TUNE1

VCLK_INV_EN DAC 6.5MHz clock phase inversion

- **0** not inverted.
- **1** inverted.

VCLK2X_INV_EN DAC 13MHz clock phase inversion. (Only 3 LSB-bit are used)

0 not inverted.

1 inverted.

VRXCALI VBIRX DAC bias current control.

0 4/4 x Iu

1 3/4 x Iu

2 2/4 x Iu

3 1/4 x Iu

4 8/4 x Iu

5 7/4 x Iu

6 6/4 x Iu

7 5/4 x Iu

6.1.2.7 Audio Front-end

MCU APB bus registers for audio are listed as followings.

0x80060200 AFE Audio Analog Gain Control Register AFE_AAG_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**APGR**				**APGL**			
Type									R/W				R/W			
Reset									0000				0000			

Set this register for analog PGA gains.

APGR audio PGA R-channel gain control

APGL audio PGA L-channel gain control

APGR [3:0] / APGL [3:0]	Gain
1111	23dB
1110	20dB
1101	17dB
1100	14dB
1011	13dB
1010	8dB
1001	5dB
1000	2dB
0111	-1dB
0110	-4dB
0101	-7dB
0100	-10dB
0011	-13dB
0010	-16dB
0001	-19dB

0000	-22dB

0x80060204 AFE Audio Analog-Circuit Control Register AFE_AAC_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			APRO_SC	ADEPOP		ABUFSELR			ABUFSELL			ACALI				
Type			R/W	R/W		R/W			R/W			R/W				
Reset			0	0		000			000			00000				

Set this register for analog circuit configuration controls.

APRO_SC Short circuit protection.
- **0** disable
- **1** enable

ADEPOP De-POP noise.
- **0** disable
- **1** enable

ABUFSELR audio buffer R-channel input selection
- **001** audio DAC R-channel output
- **010** voice DAC output
- **100** external FM R/L-channel radio output, stereo to mono
- **101** external FM R-channel radio output
- **OTHERS** reserved.

ABUFSELL audio buffer L-channel input selection
- **001** audio DAC L-channel output
- **010** voice DAC output
- **100** external FM R/L-channel radio output, stereo to mono
- **101** external FM L-channel radio output
- **OTHERS** reserved.

ACALI audio bias current control, in 2's complement format. (Only 3 LSB-bit are used)
- **0** 4/4 x Iu
- **1** 3/4 x Iu
- **2** 2/4 x Iu
- **3** 1/4 x Iu
- **4** 8/4 x Iu
- **5** 7/4 x Iu
- **6** 6/4 x Iu
- **7** 5/4 x Iu

0x80060208 AFE Audio Analog Power Down Control Register AFE_AAPDN_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											APDVCM_EN	APDN_BIAS	APDN_DACR	APDN_DACL	APDN_OUTR	APDN_OUTL
Type											R/W	R/W	R/W	R/W	R/W	R/W
Reset											0	0	0	0	0	0

Set this register to power up analog blocks. 0: power down, 1: power up. Suggested value is 00ffh.

APDNVCM_EN Power on VCM circuit (reserved)
- **0** OFF
- **1** ON

APDN_BIAS Power on bias circuit
- **0** OFF
- **1** ON

APDN_DACR Power on R-channel DAC DCT circuit
- **0** OFF
- **1** ON

APDN_DACL Power on L-channel DAC DCT circuit
- **0** OFF
- **1** ON

APDN_OUTR Power on R-channel OUT buffer block
- **0** OFF
- **1** ON

APDN_OUTL Power on L-channel OUT buffer block
- **0** OFF
- **1** ON

0x8006020C Enhanced Audio Analog Front End Control & Parameters AFE_AAC_NEW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUF_BIAS			MUX		AVCMGEN_EN	VCM_MODE	
Type									R/W			R/W		R/W	R/W	
Reset									0			00		0	0	

MT6516 ehnahced audio DAC application circuitry selection and control parameters.

BUF_BIAS Select audio buffer output stage quasi bias current.
- **00** 1X
- **01** 1.5X
- **10** 0.5X
- **11** 0.75X

MUX Mux audio DAC output to DM R/L pins. (Configure FMINR/FMINL I/O)
- **00** FM input
- **01** FM input
- **10** Left channel DAC differential output
- **11** Right channel DAC differential output

AVCMGEN_EN Power Down Buffer VCM Generation Circuit (Active Low)
- **0** OFF
- **1** ON

VCM_MODE Change output buffer common mode generation circuitry.
- **0** New VCM circuitry
- **1** Old VCM circuitry

0x80060210 AFE Audio Analog-Circuit Control Register 1 AFE_AAC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							ACLK2X_INV_EN	ACLK_INV_EN	ACLK_SEL_EN				ABIAS_DCT_EN			
Type							R/W	R/W	R/W				R/W			
Reset							0	0	0000				0000			

Set this register for analog circuit configuration controls.

ABIAS_DAC_EN BIAS current tunning register for DCT.

0~15 corresponds to 8/8,7/8,6/8,5/8,4/8,3/8,2/8,1/8,16/8,15/8,14/8,13/8,12/8,11/8,10/8,9/8

ACLK_SEL_EN Non-overlap clock timing tunning register

[3:2] clock delay fine tune.

[1:0] non-overlapping clock fine tune.

ACLK_INV_EN DAC 6.5MHz clock phase inversion

0 not inverted.

1 inverted.

ACLK2X_INV_EN DAC 13MHz clock phase inversion.

0 not inverted.

1 inverted.

6.1.2.8 Register setting path

0x8301000C Switch the register configuring path CCI_WR_PATH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												AD_INF_PATH	PLL_WR_PATH	MODEM_WR_PATH	VBI_WR_PATH	ABI_WR_PATH
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

WR_PATH

0 Switch the register setting to MCU side

1 Switch the register setting to manually control by TRACE32 through JTAG

The bit is to facilitate ACD members for verifying purpose; the hardware supports write path switching, without being disturbed by existing MCU load. However, when with manually control, all register addresses are offset by 0x1000. For example, MCU configures AFE_AAC_NEW through the address 0x8301020c, while the manually control path take effect when configuring 0x8301120c. Notice that before finishing manual control, the register must be reset to be 0. The modem part includes BBRX, BBTX, APC, AFC, WBRX, TXVGA, VBIAS and AUXADC.

AD_INF_PATH The register bit decides the input/output path of the mixed-mode module. For ABI and VBI, it can be configured to feed the pattern from AFE or from CHIP I/O (shared with A_FUNC_MODE). For BBTX, TXVGA, VBIAS, APC, and AFC, the input selection interface is divided at either MIX_DIG or CHIP I/O (also shared with A_FUNC_MODE). As for the BBRX and WBRX, the output pattern can be bypass to CHIPIO with this register bit being true. The bit is for convenient debug-use in normal mode, such that the data pattern can be observed or be feed-in by external device, while control register

setting still comes from the chip internally(By use of JTAG). It should be notice that this special debug mode should be accompanied by proper setting of GPIO, which decides the PAD OE when in normal function.

0 data pattern comes from chip internally, and the output data cannot be bypassed to chip I/O

1 analog debug mode in normal function

6.1.2.9 Reserved

Some registers are reserved for further extensions.

0x83010900h Reserved 1 Analog Circuit Control Register 0 RES1_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010904h Reserved 1 Analog Circuit Control Register 1 RES1_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010A00h Reserved 2 Analog Circuit Control Register 0 RES2_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010A04h Reserved 2 Analog Circuit Control Register 1 RES2_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010B00h Reserved 3 Analog Circuit Control Register 0 RES3_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2009 MediaTek Inc.

0x83010B04h Reserved 3 Analog Circuit Control Register 1 RES3_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010C00h Reserved 4 Analog Circuit Control Register 0 RES4_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010C04h Reserved 4 Analog Circuit Control Register 1 RES4_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010D00h Reserved 5 Analog Circuit Control Register 0 RES5_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010D04h Reserved 5 Analog Circuit Control Register 1 RES5_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010E00h Reserved 6 Analog Circuit Control Register 0 RES6_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2009 MediaTek Inc.

0x83010E04h Reserved 6 Analog Circuit Control Register 1 RES6_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010F00h Reserved 7 Analog Circuit Control Register 0 RES7_AC_CON0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x83010F04h Reserved 7 Analog Circuit Control Register 1 RES7_AC_CON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.1.3 Programming Guide

6.1.3.1 BBRX Register Setup

The register used to control analog base-band receiver is BBRX_AC_CON.

6.1.3.1.1 Programmable Biasing Current

To maximize the yield in modern digital process, the receiver features providing 5-bit 32-level programmable current to bias internal analog blocks. The 5-bits registers CALBIAS [4:0] is coded with 2’s complement format.

6.1.3.1.2 Offset / Gain Calibration

The base-band downlink receiver (RX), together with the base-band uplink transmitter (TX) introduced in the next section, provides necessary analog hardware for DSP algorithm to correct the mismatch and offset error. The connection for measurement of both RX/TX mismatch and gain error is shown in **Figure 150**, and the corresponding calibration procedure is described below.

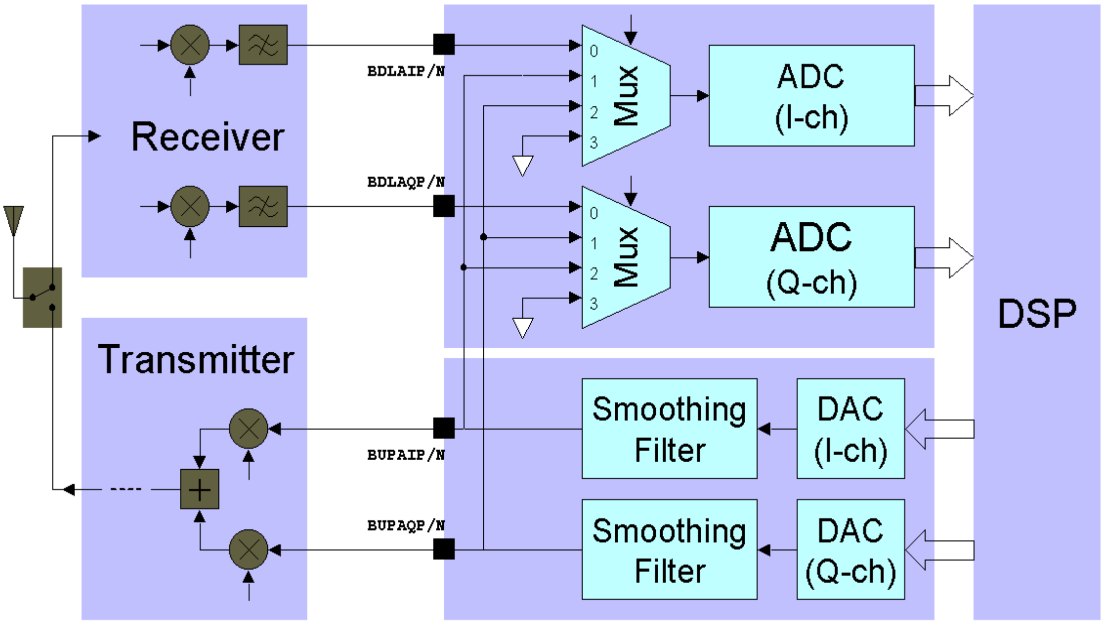


Figure 150 Base-band A/D and D/A Offset and Gain Calibration

6.1.3.1.3 Downlink RX Offset Error Calibration

The RX offset measurement is achieved by selecting grounded input to A/D converter (set ISEL [1:0] ='11' and QSEL [1:0] ='11' to select channel 3 of the analog input multiplexer, as shown in **Figure** . The output of the ADC is sent to DSP for further offset cancellation. The offset cancellation accuracy depends on the number of samples being converted. That is, more accurate measurement can be obtained by collecting more samples followed by averaging algorithm.

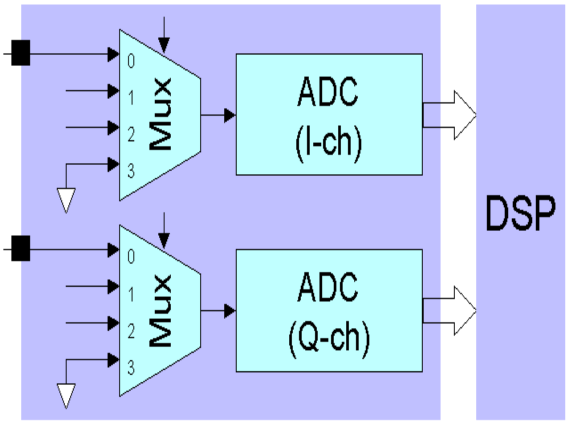


Figure 5 Downlink ADC Offset Error Measurement

6.1.3.1.4 Downlink RX and Uplink TX Gain Error Calibration

To measure the gain mismatch error, both I/Q uplink TXs should be programmed to produce full-scale pure sinusoidal waves output. Such signals are then fed to downlink RX for A/D conversion, in the following two steps.

A. The uplink (both in GSM and WCDMA mode) I-channel output are connected to the downlink I-channel input, and the uplink Q-channel output are connected to the downlink Q-channel input. This can be achieved by setting ISEL [1:0] ='01' and QSEL [1:0] ='01' (shown in **Figure** 151 (A))..

B. The uplink (both in GSM and WCDMA mode) I-channel output are then connected to the downlink Q-channel input, and the uplink Q-channel output are connected to the downlink I-channel input. This can be achieved by setting ISEL [1:0] ='10' and QSEL [1:0] ='10' (shown in **Figure** 151 (B)).

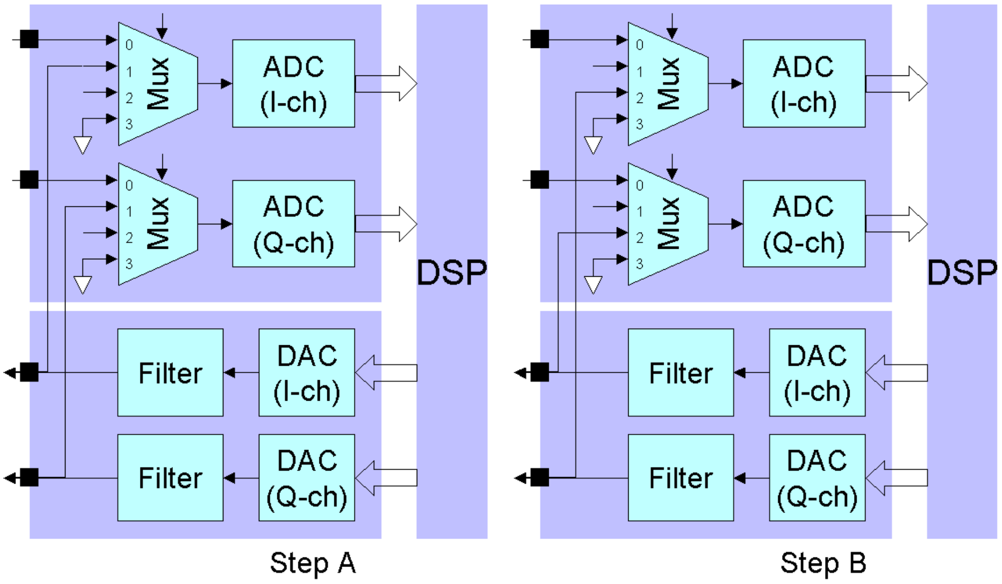


Figure 151 Downlink RX and Up-link TX Gain Mismatch Measurement (A) I/Q TX connect to I/Q RX (B) I/Q TX connect to Q/I RX

Once above successive procedures are completed, RX/TX gain mismatch could be easily obtained because the amplitude mismatch on RX digitized result in step A and B is the sum and difference of RX and TX gain mismatch, respectively.

The gain error of the downlink RX can be corrected in the DSP section and the uplink TX gain error can be corrected by the gain trimming facility that TX block provide.

6.1.3.1.5 Uplink TX Offset Error Calibration

Once the offset of the downlink RX is known and corrected, the offset of the uplink TX alone could be easily estimated. The offset error of TX should be corrected in the digital domain by means of the programmable feature of the digital GMSK modulator.

Finally, it is important that above three calibration procedures should be exercised in order, that is, correct the RX offset first, then RX/TX gain mismatch, and finally TX offset. This is owing to that analog gain calibration in TX will affect its offset, while the digital offset correction has no effect on gain.

6.1.3.2 BBTX Register Setup

The register used to control analog base-band transmitter is BBTX_AC_CON0~4.

 © 2009 MediaTek Inc.

6.1.3.2.1 Output Gain Control

The output swing of the uplink transmitter is controlled by register GAIN [2:0] (or WB_GAIN[2:0]) coded in 2's complement with about 2dB step. When TRIMI [3:0] (WB_TRIMI[3:0]) / TRIMQ [3:0] (WB_TRIMQ[3:0]) = 4'b0000 the swing is listed in **Table** 141, defined to be the difference between positive and negative output signal.

GAIN [2:0] / WB_GAIN[2:0]	Output Swing	For AVDD=2.8 (V)
+3 (011)	AVDD*0.526 (+3.36 dB)	1.46
+2 (010)	AVDD*0.462 (+2.24 dB)	1.29
+1 (001)	AVDD*0.406 (+1.12 dB)	1.14
+0 (000)	AVDD*0.357 (+0.00 dB)	1
-1 (111)	AVDD*0.314 (-1.12 dB)	0.88
-2 (110)	AVDD*0.278 (-2.24 dB)	0.78
-3 (101)	AVDD*0.243 (-3.36 dB)	0.68
-4 (100)	AVDD*0.214 (-4.48 dB)	0.6

Table ***141 Output Swing Control Table***

6.1.3.2.2 Output Gain Trimming

I/Q channels can also be trimmed in GSM/GPRS or WCDMA mode separately to compensate gain mismatch in the base-band transmitter or the whole transmission path including RF module. The gain trimming is adjusted in 16 steps spread from –0.96dB to +0.84dB (**Table** 142), compared to the full-scale range set by GAIN [2:0]=3'b000.

TRIMI [3:0] / TRIMQ [3:0] WB_TRIMI[3:0]/WB_TRIMQ[3:0]	Gain Step (dB)
+7 (0111)	0.84
+6 (0110)	0.72
+5 (0101)	0.60
+4 (0100)	0.48
+3 (0011)	0.36
+2 (0010)	0.24
+1 (0001)	0.12
+0 (0000)	0.00
-1 (1111)	-0.12
-2 (1110)	-0.24
-3 (1101)	-0.36
-4 (1100)	-0.48

© 2009 MediaTek Inc.

-5 (1011)	-0.60
-6 (1010)	-0.72
-7 (1001)	-0.84
-8 (1000)	-0.96

*Table **142 Gain Trimming Control Table***

6.1.3.2.3 Output Common-Mode Voltage

The output common-mode voltage is controlled by CMV [2:0] (WB_CMV[2:0]) with about 0.08*AVDD step, as listed in the following table.

CMV [2:0] / WB_CMV[2:0]	Common-Mode Voltage
+3 (011)	AVDD*0.70
+2 (010)	AVDD*0.62
+1 (001)	AVDD*0.54
+0 (000)	AVDD*0.50
-1 (111)	AVDD*0.46
-2 (110)	AVDD*0.42
-3 (101)	AVDD*0.38
-4 (100)	AVDD*0.30

Table 143 Output Common-Mode Voltage Control Table

6.1.3.2.4 Programmable Biasing Current

The transmitter features providing 4-bit 9-level programmable current to bias internal analog blocks. It can control bias current from 0.5x to 2x.

6.1.3.2.5 Smoothing Filter Characteristic

The 3rd–order Butterworth smoothing filter is used to suppress the image at DAC output: it provides more than 60dB attenuation at the 4.33MHz sampling frequency. To tackle with the digital process component variation, programmable cutoff frequency control bits CALRCSEL [2:0] are included. User can directly change the filter cut-off frequency by different CALRCSEL value (coded with 2's complement format and with a default value 0). In addition, an internal calibration process is provided, by setting START CALRC to high and CALRCCNT to an appropriate value (default is 11). After the calibration process, the filter cut-off frequency is calibrated to 350kHz +/- 50 kHz and a new CALRCOUT value is stored in the register. During the calibration process, the output of the cell is high-impedance. After this calibration, cut-off frequency of the filter in WCDMA mode is also set to its applicable value.

6.1.3.3 AFC-DAC Register Setup

The register used to control the APC DAC is AFC_AC_CON, which providing 9 different programmable current to bias internal analog blocks. Note that the 5-bits registers CALI [4:0] is NOT coded with 2's complement format.

There is another 1-bit register TGSEL to control the sampling phase, either rising edge or falling edge, for DAC code.

6.1.3.4 APC-DAC Register Setup

The register used to control the APC DAC is AFC_AC_CON, which providing 9 different programmable current to bias internal analog blocks. Note that the 5-bits registers CALI [4:0] is NOT coded with 2's complement format.

There is another 1-bit register TGSEL to control the sampling phase, either rising edge or falling edge, for DAC code.

6.1.3.5 Auxiliary A/D Conversion Register Setup

The register used to control the Aux-ADC is AUX_AC_CON. For this register, which providing 9 different programmable current to bias internal analog blocks..

6.1.3.6 Voice-band Blocks Register Setup

The registers used to control AMB are AFE_VAG_CON, AFE_VAC_CON0, AFE_VAC_CON1, and AFE_VAPDN_CON. For these registers, please refer to chapter "Analog Front End & Analog Blocks"

6.1.3.6.1 Reference Circuit

The voice-band blocks include internal bias circuits, a differential bandgap voltage reference circuit and a differential microphone bias circuit. Internal bias current could be calibrated by varying VCALI[4:0] (coded with 2's complement format).

For proper operation, there should be an external 1uF capacitor connected between differential output pins AU_VCM_PO and AU_VCM_NO. The following table illustrates typical 0dBm0 voltage when uplink/downlink programmable gains are unity. For other gain setting, 0dBm0 reference level should be scaled accordingly.

Symbol	Parameter	Min	Typical	Max	Unit
$V_{0dBm0,UP}$	0dBm0 Voltage for Uplink Path, Applied Differentially Between Positive and Negative Microphone Input Pins		0.2V		V-rms
$V_{0dBm0,Dn}$	0dBm0 voltage for Downlink Path, Appeared Differentially Between Positive and Negative Power Amplifier Output Pins		0.6V		V-rms

Table 144 ***0dBm0 reference level for unity uplink/downlink gain***

The microphone bias circuit generates a single-ended output voltage on AU_MICBIAS_P for external electret type microphone. Typical output voltage is 1.9 V. The max current supplied by microphone bias circuit is 2mA.

6.1.3.6.2 Uplink Path

Uplink path of voice-band blocks includes an uplink programmable gain amplifier and a sigma-delta modulator.

 © 2009 MediaTek Inc.

6.1.3.6.3 Uplink Programmable Gain Amplifier

Input to the PGA is a multiplexer controlled by VCFG [3:0], as described in the following table. In normal operation, both input AC and DC coupling are feasible for attenuation the input signal (gain <= 0dB). However, only AC coupling is suggested if amplification of input signal is desired (gain>=0dB).

Control Signal	Function	Descriptions
VCFG [0]	Input Selector	0: Input 0 (From AU_VIN0_P / AU_VIN0_N) is selected 1: Input 1 (From AU_VIN1_P / AU_VIN1_N) is selected
VCFG [1]	Input Selector	1: Input FM (From AU_FMINL / AU_FMINR) is selected
VCFG [2]	Coupling Mode	0: AC Coupling 1: DC Coupling
VCFG [3]	Gain Mode	0: Amplification Mode (gain range -20~43 dB) 1: Bypass Mode

Table 145 ***Uplink PGA input configuration setting***

The PGA itself provides programmable gain (through VUPG [5:0]) with step of 1dB, as listed in the following table.

VUPG [5:0]	Gain	VUPG [5:0]	Gain
111111	43 dB	011111	11dB
111110	42 dB	011110	10dB
111101	41 dB	011101	9dB
111100	40 dB	011100	8dB
111011	39 dB	011011	7dB
111010	38 dB	011010	6dB
111001	37 dB	011001	5dB
111000	36 dB	011000	4dB
110111	35 dB	010111	3dB
110110	34 dB	010110	2dB
110101	33 dB	010101	1dB
110100	32 dB	010100	0dB
110011	31 dB	010011	-1dB
110010	30 dB	010010	-2dB
110001	29 dB	010001	-3dB
110000	28 dB	010000	-4dB
101111	27 dB	001111	-5dB
101110	26 dB	001110	-6dB
101101	25 dB	001101	-7dB
101100	24 dB	001100	-8dB

101011	23 dB	001011	-9dB
101010	22 dB	001010	-10dB
101001	21 dB	001001	-11dB
101000	20dB	001000	-12dB
100111	19 dB	000111	-13dB
100110	18 dB	000110	-14dB
100101	17 dB	000101	-15dB
100100	16 dB	000100	-16dB
100011	15 dB	000011	-17dB
100010	14 dB	000010	-18dB
100001	13 dB	000001	-19dB
100000	12dB	000000	-20 dB

Table 146 ***Uplink PGA gain setting when inputs apply at microphone (VUPG [5:0])***

The following table illustrates typically the 0dBm0 voltage applied at the microphone inputs, differentially, for several gain settings.

VCFG [3] ='0'		VCFG [3] ='1' (only valid for input 1)	
VUPG [5:0]	0dBm0 (V-rms)	VUPG [5:0]	0dBm0 (V-rms)
111100	2mV	XXXXXX	0.2V
101000	20mV		
100000	50mV		
010100	0.2V		

Table 147 ***0dBm0 voltage at microphone input pins***

6.1.3.6.4 Sigma-Delta Modulator

Analog-to-digital conversion in uplink path is made with a second-order sigma-delta modulator (SDM) whose sampling rate is 4096kHz. Output signals are coded in either one-bit or RSD format, optionally controlled by VRSDON register.

For test purpose, one can set VADCINMODE to HI to form a look-back path from downlink DAC output to SDM input. The default value of VADCINMODE is zero.

6.1.3.6.5 Downlink Path

Downlink path of voice-band blocks includes a digital to analog converter (DAC) and two programmable output power amplifiers.

6.1.3.6.6 Digital to Analog Converter

The DAC converts input bit-stream to analog signal by sampling rate of 4096kHz. . Besides, it performs a 2nd-order 40kHz butterworth filtering. The DAC receives input signals from MT6516 DSP by set VDACINMODE = 0. It can also take inputs from SDM output by setting VDACINMODE = 1.

6.1.3.6.7 Downlink Programmable Power Amplifier

Voice-band analog blocks include two identical output power amplifiers with programmable gain. Amplifier 0 and amplifier 1 can be configured to either differential or single-ended mode by adjusting VDSEND [0] and VDSEND [1], respectively. In single-ended mode, when VDSEND[0] =1, output signal is present at AU_VOUT0_P pin respect to ground. Same as VDSEND[1] for AU_VOUT1_P pin.

For the amplifier itself, programmable gain setting is described in the following table.

VDPG0 [3:0] / VDPG1 [3:0]	Gain
1111	8dB
1110	6dB
1101	4dB
1100	2dB
1011	0dB
1010	-2dB
1001	-4dB
1000	-6dB
0111	-8dB
0110	-10dB
0101	-12dB
0100	-14dB
0011	-16dB
0010	-18dB
0001	-20dB
0000	-22dB

Table 148 ***Downlink power amplifier gain setting***

Control signal VFLOAT, when set to 'HI', is used to make output nodes totally floating in power down mode. If VFLOAT is set to 'LOW" in power down mode, there will be a resistor of 50k ohm (typical) between AU_VOUT0_P and AU_VOUT0_N, as well as between AU_VOUT0_P and AU_VOUT0_N.

The amplifiers deliver signal power to drive external earphone. The minimum resistive load is 28 ohm and the upper limit of the output current is 50mA. On the basis that 3.14dBm0 digital input signal into downlink path produces DAC output differential voltage of 0.87V-rms (typical), the following table illustrates the power amplifier output signal level (in V-rms) and signal power for an external 32 ohm resistive load.

VDPG	Output Signal Level (V-rms)	Output Signal Power (mW / dBm)
0010	0.11	0.37/-4.3
0110	0.27	2.28/3.6
1010	0.69	14.8/11.7
1110	1.74	94.6/19.8

Table 149 ***Output signal level/power for 3.14dBm0 input. External resistive load = 32 ohm***

The following table illustrates the output signal level and power for different resistive load when VDPG =1110.

RLOAD	Output Signal Level (V-rms)	Output Signal Power (mW / dBm)
30	1.74	101/20
100	1.74	30.3/14.8
600	1.74	5/7

Table 150 ***Output signal level/power for 3.14dBm0 input, VDPG =1110***

6.1.3.6.8 Power Down Control

Each block inside audio mixed-signal blocks features dedicated power-down control, as illustrated in the following table.

Control Signal	Descriptions
VPDN_BIAS	Power Down Reference Circuits (Active Low)
VPDN_LNA	Power Down Uplink PGA (Active Low)
VPDN_ADC	Power Down Uplink SDM (Active Low)
VPDN_DAC	Power Down DAC (Active Low)
VPDN_OUT0	Power Down Downlink Power Amp 0 (Active Low)
VPDN_OUT1	Power Down Downlink Power Amp 1 (Active Low)

Table 151 ***Voice-band blocks power down control***

6.1.3.7 Audio-band Blocks Register Setup

The registers used to control audio blocks are AFE_AAG_CON, AFE_AAC_CON, AFE_AAC_CON, AFE_AAC_NEW_CON, AFE_AAC_CON1 and AFE_AAPDN_CON.

6.1.3.7.1 Output Gain Control

Audio blocks include stereo audio DACs and programmable output power amplifiers. The DACs convert input bit-stream to analog signal by sampling rate of 6500kHz by sample-rate converter. The two identical output power amplifiers with programmable gain are designed to driving external AC-coupled single-end speaker. The minimum resistor load is 16 ohm. The programmable gain setting, controlled by APGR[] and APGL[], is the same as that of the voice-band amplifiers.

Unlike voice signals, 0dBFS defines the full-scale audio signals amplitude. The following table illustrates the power amplifier output signal level (in V-rms) and signal power for an external 16 ohm resistive load.

APGR[]/ APGL[]	Output Signal Level (V-rms)	Output Signal Power (mW / dBm)
0010	0.055	0.19/-7.2
0110	0.135	1.14/0.6

 © 2009 MediaTek Inc.

1010	0.345	7.44/8.7
1110	0.87	47.3/16.7

Table 152 ***Output signal level/power for 0dBFS input. External resistive load = 16 ohm***

6.1.3.7.2 Mute Function and Power Down Control

By setting AMUTER (AMUTEL) to high, right (Left) channel output will be muted.

Each block inside audio mixed-signal blocks features dedicated power-down control, as illustrated in the following table.

Control Signal	Descriptions
APDN_BIAS	Power Down Audio Bias Circuits (Active Low)
APDN_DACL	Power Down L-Channel DAC (Active Low)
APDN_DACR	Power Down R-Channel DAC (Active Low)
APDN_OUTL	Power Down L-Channel Audio Amplifier (Active Low)
APDN_OUTR	Power Down R-Channel Audio Amplifier (Active Low)
AVCMGEN_EN	Power Down Buffer VCM Generation Circuit (Active Low)

Table 153 ***Audio-band blocks power down control***

6.1.3.8 Multiplexers for Audio and Voice Amplifiers

The audio/voice amplifiers feature accepting signals from various signal sources including AU_FMINR/AU_FMINL pins, that aimed to receive stereo AM/FM signal from external radio chip:

1) Voice-band amplifier accepts signals from voice DAC output only.
2) Audio left/right channel amplifiers receive signals from either voice DAC, audio DAC, or AM/FM radio input pins (controlled by registers ABUFSELL[] and ABUFSELR[]), too. Left and right channel amplifiers will produce identical output waveforms when receiving mono signals from voice DAC.

6.1.3.9 Preferred Microphone and Earphone Connections

In this section, preferred microphone and earphone connections are discussed.

Differential connection of microphone is shown below. This is the application circuits compatible with previous products C1 and Rin form an AC coupling and high-pass network. C1*Rin should be chosen such that the in-band signal will not be attenuated too much. For differential minimum resistance of 13k ohm, minimum value of C1 is 170nF for less than 1dB attenuation at 300Hz. R2 is determined by microphone sensitivity. C2 and R2 form another low-pass filter to filtering noise coming from microphone bias pins. Pole frequency less than 50Hz is recommended.

© 2009 MediaTek Inc.

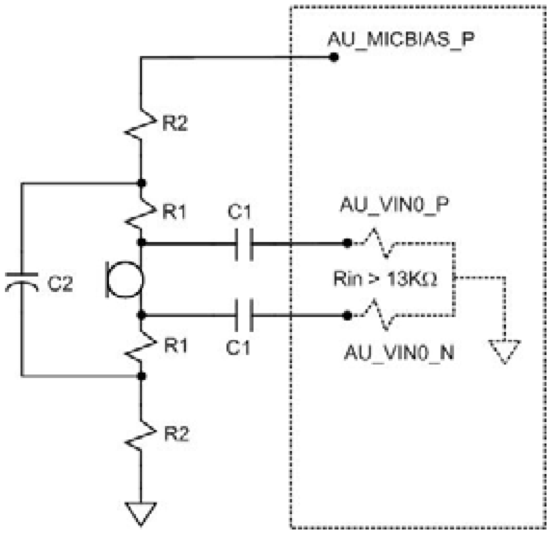


Figure 6 Differential Microphone Connection

Another suggested connection method of microphone is shown below. R1 is chosen based on microphone sensitivity requirement. C1 and Rin form an AC coupling and high-pass network. R2 needs proper adjustment to obtain the best noise performance on the voice uplink input terminals.

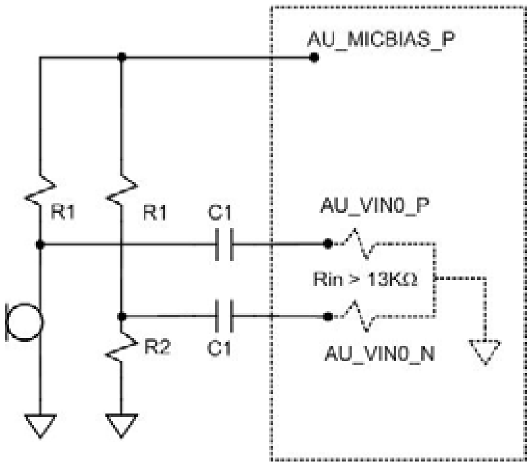


Figure 7 Suggested Microphone Connection

For earphone, both connections can be used. The application circuit shown in Figure 7 is highly recommended to achieve the better performance.

6.1.3.10 Clock Squarer Register Setup

The register used to control clock squarer is CLK_CON. For this register, please refer to chapter "Clocks"

. The register used to control the CLKSQ are CLKSQ_DIV2_PWD and CLKSQ_DIV2_SEL, in which CLKSQ_DIV2_SEL is a 1bit register that is used to provide correct input frequency for PLL under different CLKSQ input frequency. and CLKSQ_DIV2_PWD is also 1bit register to control divider by 2 circuit in CLKSQ to turn on or not . Please refer to the register table for detail

 © 2009 MediaTek Inc.

6.1.3.11 Phase-Locked Loop Register Setup

For registers control the PLL, please refer to chapter “Clocks” and “Software Power Down Control”

6.1.3.11.1 Frequency Setup

Camera PLL is the only one PLL that is frequency adjustable. The register used to control the CPLL output frequency is CPLL_FBDIV, which providing 6-bit to control PLL feedback divider. Please refer to the register table for detail

6.1.3.11.2 Programmable Biasing Current

The PLLs feature providing 5-bit 32-level programmable current to bias internal analog blocks. The 5-bits registers CALI [4:0] is coded with 2’s complement format.

6.1.3.12 32-khz Crystal Oscillator Register Setup

For registers that control the oscillator, please refer to chapter “Real Time Clock” and “Software Power Down Control”.
XOSCCALI[4:0] is the calibration control registers of the bias current, and is coded with 2’s complement format.

CL is the parallel combination of C1 and C2 in the block diagram.

6.2 Clocks

There are two major time bases in the MT6516. For the faster one is the 13 MHz clock originating from an off-chip temperature-compensated voltage controlled oscillator (TCVCXO) that can be either 13MHz or 26MHz. This signal is the input from the SYSCLK pad then is converted to the square-wave signal. The other time base is the 32768 Hz clock generated by an on-chip oscillator connected to an external crystal. **Figure 152** shows the clock sources as well as their utilizations inside the chip.

© 2009 MediaTek Inc.

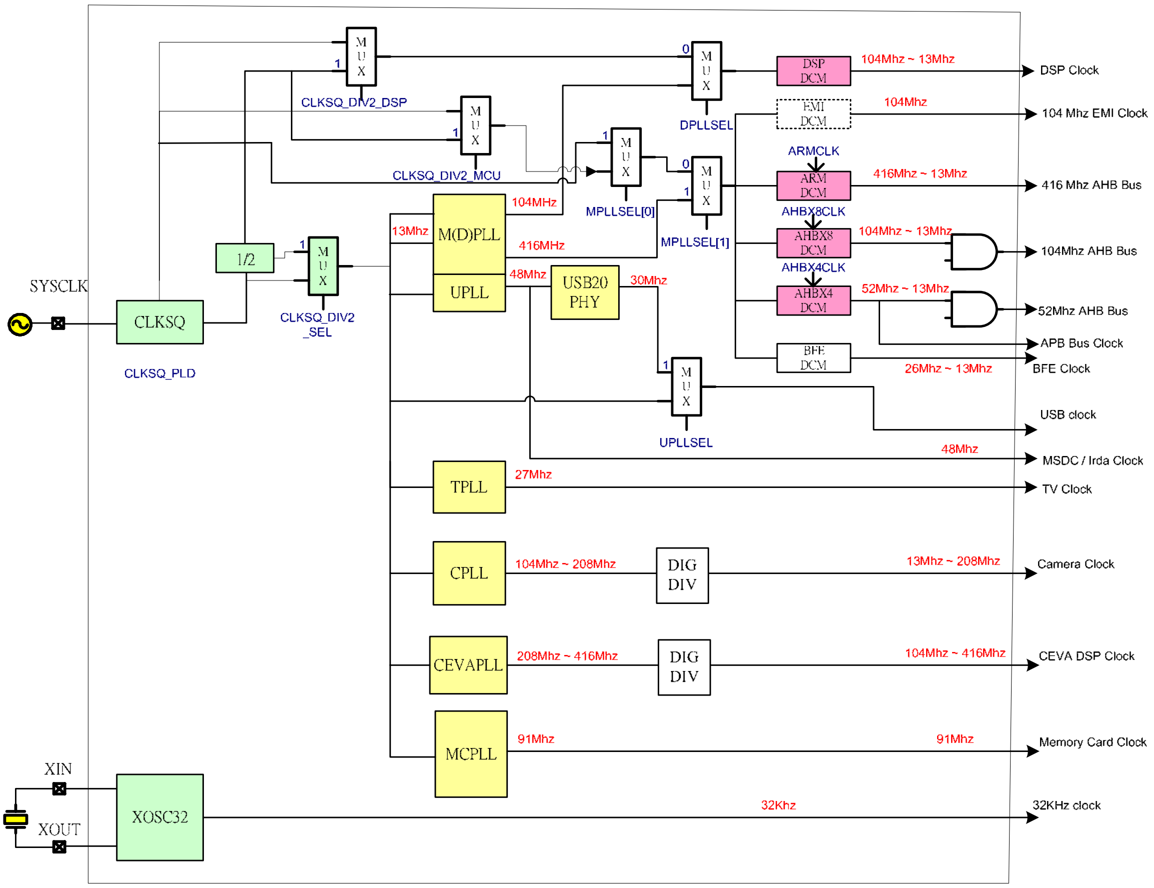


Figure 152 ***Clock distributions inside the MT6516***

6.2.1 32.768 KHz Time Base

The 32768 Hz clock is always running. It's mainly used as the time base of the Real Time Clock (RTC) module, which maintains time and date with counters. Therefore, both the 32768Hz oscillator and the RTC module is powered by separate voltage supplies that shall not be powered down when the other supplies do. In low power mode, the 13 MHz time base is turned off, so the 32768 Hz clock shall be employed to update the critical TDMA timer and Watchdog Timer. This time base is also used to clocks the keypad scanner logic.

6.2.2 13 MHz Time Base

One 1/2-dividers for PLL existing to allow using 26 or 13 MHz TCVCXO.

One phase-locked loops (MPLL) to generate 104Mhz and 416Mhz, for two clocks, DSP_CLOCK and MCU_CLOCK, respectively. These two primary clocks then feed to DSP Clock Domain and MCU Clock Domain, respectively. Another phase-locked loop (UPLL) is to generate 48Mhz for the two clocks, USB_CLOCK and IRDA_CLOCK, respectively.

The PLL require no off-chip components for operations and can be turn off in order to save power. After power-on, the PLLs are off by default and the source clock signal is selected through multiplexers. The

software shall take cares of the PLL lock time while changing the clock selections. The PLL and usages are listed below.

UPLL supplies two clock source

USB system clock, *USB_CLOCK*. The 48MHz is sent to USB module for its operation.

IRDA system clock, *IRDA_CLOCK*. The 48MHz is sent to IRDA module for its operation.

MPLL supplies two clock source

DSP system clock, *DSP_CLOCK*. The outputted 182MHz clock is connected to DSP DCM (dynamic clock manager) for dynamically adjusting clock rate by digital clock divider.

MCU system clock, *MCU_CLOCK*, which paces the operations of the MCU cores, MCU memory system, and MCU peripherals as well. The outputted 416MHz clock is connected to ARM DCM and AHB DCM for dynamically adjusting clock rate by digital clock divider. The usage of DCM is described in MCUCLK_CON registers of CONFIG.

CPLL supplies one clock source

Camera clock, *CAM_CLOCK*. The outputted 104~208MHz clock is connected to a digital clock divider to get 13~208MHz clock for camera system.

TPLL supplies one clock source

TV system clock, *TV_CLOCK*. The outputted 27MHz clock is connected to TV-OUT DAC.

CEVAPLL supplies one clock source

CEVA DSP system clock, *CEVA DSP_CLOCK*. The outputted 104MHz~416MHz clock is connected to CEVA DSP Domain.

MCPLL supplies one clock source

Memory Card system clock, *MC CLOCK*. The outputted 91MHz clock is connected to Memory card control unit.

Note that PLL need some time to become stable after being powered up. The software shall take cares of the PLL lock time before switching them to the proper frequency. Usually, a software loop longer than the PLL lock time is employed to deal with the problem.

For power management, the MCU software program may stop MCU Clock by setting the Sleep Control Register. Any interrupt requests to MCU can pause the sleep mode, and thus MCU return to the running mode.

AHB also can be stop by setting the Sleep Control Register. However the behavior of AHB in sleep mode is a little different from that of MCU. After entering Sleep Mode, it can be temporarily waken up by any “hreq” (bus request), and then goes back to sleep automatically after all “hreqs” de-assert. Any transactions can take place as usual in sleep mode, and it can save power while there is no transaction on it. However the penalty is losing a little system efficiency for switching on and off bus clock, but the impact is small.

6.2.3 Dynamic Clock Switch of MCU Clock

Dynamic Clock Manager is implemented to allow MCU and DSP switching clock dynamically without any jitter, and enabling signal drift, and system can operate stably during any clock rate switch.

Before switching to PLL clocks, the clock from PLL DIV2 will feed through dynamic clock manager (DCM) directly. That means if PLL DIV2 is enabled, the internal clock rate is the half of SYSCLK. Contrarily, the internal clock rate is identical to SYSCLK.

 © 2009 MediaTek Inc.

However, the settings of some hardware modules are required to be changed before or after clock rate change. Software has the responsibility to change them at proper timing. The following table is list of hardware modules needed to be changed their setting during clock rate change.

Module Name	Programming Sequence
NAND	1. Low clock speed -> high clock speed Changing wait state before clock change. New wait state will not take effect until current EMI access is complete. Software should insert a period of time before switching clock. 2. High clock speed -> low clock speed Changing wait state after clock change.
LCD	Change wait state while LCD in IDLE state.

Table 154 ***Programming sequence during clock switch***

6.2.4 Standard PLL Power-on sequence

```
*PDN_CON   = 0x1e     // power-on MPLL(DPLL), UPLL
*CEVAPLL2  = 0x1f     // power-on CEVAPLL
*CLK_CON   = 0x83     // switch to 13MHz for PLL input frequency
// After power-on PLL.....
*UPLL = 0x0080; // reset UPLL
*MPLL = 0x0080; // reset MPLL
*CEVAPLL = 0x0800; // reset CEVAPLL
*UPLL = 0x0000; // release UPLL reset
*MPLL = 0x0000; // release MPLL reset
*CEVAPLL = 0x0000; // release CEVAPLL reset

for (i=0;i<200;i++);
*CLK_CON = 0x00f3; // select PLL outputs
```

6.2.5 Register Definitions

Table2 list Clock and PLL Control Registers and their address mapping to the MCU address space. The Clock and PLL Control register addresses starts from the base address 0x80060000.

Register Address	Register Function	Acronym
0x80060010	Power-down Control Register	**PDN_CON**
0x80060014	Clock Control Register	**CLK_CON**
0x80060018	Reserved.	**-**
0x8006001C	Reserved.	**-**
0x80060020	MCU (DSP) PLL Control Register	**MPLL**
0x80060024	MCU (DSP) PLL Control Register 2	**MPLL2**
0x80060028	Reserved.	**-**
0x8006002C	Reserved.	**-**
0x80060030	USB PLL Control Register	**UPLL**
0x80060034	USB PLL Control Register 2	**UPLL2**

 © 2009 MediaTek Inc.

0x80060038	Camera PLL Control Register	**CPLL**
0x8006003C	Camera PLL Control Register 2	**CPLL2**
0x80060040	TV PLL Control Register	**TPLL**
0x80060044	TV PLL Control Register 2	**TPLL2**
0x80060048	Camera PLL Control Register 3	**CPLL3**
0x8006004C	PLL Reserved Control Register 0	**PLL_RES_CON0**
0x80060050	PLL Bias Control Register	**PLL_BIAS**
0x80060054	Reserved.	**-**
0x80060058	Memory card PLL Control Register	**MCPLL**
0x8006005c	Memory card PLL Control Register 2	**MCPLL2**
0x80060060	CEVA PLL Control Register	**CEVAPLL**
0x80060064	CEVA PLL Control Register 2	**CEVAPLL2**
0x80060070	PLL IDN Control Register	**PLL_IDN**
0x8006007C	XOSC32 Analog Circuit Control Register	**XOSC32_AC_CON**

Table 155 ***Clock/PLL Control Registers***

0x80060010 Power-down control PDN_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**UPLL_PWDB**	**DPLL_PWDB**	**MPLL_PWDB**	**CLKSQ_PWDB**	**CLKSQ_DIV2_PWD**
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	1	0

CLKSQ_DIV2_PWD Control CLKSQ divide-by-2 power-down (Active-High)

0 Power-on

1 Power-down

CLKSQ_PWDB Control CLKSQ power-down. (Active-Low) (default ON)

0 Power-down

1 Power-on

MPLL_PWDB Control MCU PLL power-down for MCU clock (Active-Low)

0 Power-down

1 Power-on

DPLL_PWDB Control DSP PLL power-down for DSP clock (Active-Low)

0 Power-down

1 Power-on

UPLL_PWDB Control USB PLL power-down for USB clock and MPLL clock source. (Active-Low)

0 Power-down

1 Power-on

0x80060014 Clock Control Register CLK_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CLKSQ_BYP								SRCCLK	UPLLSEL	DPLLSEL	MPLLSEL		CLKSQ_PLD	CLKSQ_DIV2_MCU	CLKSQ_DIV2_DSP
Type	R/W								R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0								1	0	0	00		0	0	0

CLKSQ_DIV2_DSP Control the clock divider for DSP clock domain

0 Divider bypassed

1 Divider not bypassed

CLKSQ_DIV2_MCU Control the x2 clock divider for MCU clock domain

0 Divider bypassed

1 Divider not bypassed

CLKSQ_PLD Pull Down Control (for DEBUG only)

0 Disable

1 Enables

MPLLSEL Select MCU Clock source. Using this mux to gate out unstable clock output from PLL after system boot up

00 PLL bypassed, using CLK from CLKSQ, default value after chip power up.

01 PLL bypassed, using CLK from SYSCLK

10 Using PLL Clock for MCU

11 Reserved

DPLLSEL Select DSP Clock source. Using this mux to gate out unstable clock output from PLL after system boot up

0 PLL bypassed, using CLK from CLKSQ

1 Using PLL Clock for DSP

UPLLSEL Select USB Clock source. Using this mux to gate out unstable clock output from PLL after system boot up

0 PLL bypassed, using CLK from CLKSQ

1 Using PLL Clock for USB

SRCCLK off-chip temperature-compensated voltage controlled oscillator (TCVCXO) frequency identifier.

0 13MHz

1 26MHz

CLKSQ_BYP CLKSQ bypass test mode (for DEBUG only)

0 Disable

1 Enables

0x80060020h MCU (DSP) PLL Control Register MPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPLL_VCOCALOK	MPLL_CP_P							MPLL_RST				MPLL_CM			
Type	R	R/W							R/W				R/W			

Reset	-	000							0				0000			

MPLL_VCOCALOK MPLL VCOBAND Calibration OK

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective when MPLL_VCOCAL_EN (in register MPLL2) is high.

MPLL_CP_P MPLL BANDWIDTH CONTROL (for DEBUG only)

MPLL_RST Reset Control of MPLL

- **0** Normal Operation
- **1** Reset the PLL

MPLL_CM MPLL Capacitor multiplier ratio (for DEBUG only)

0x80060024h MCU (DSP PLL) Control Register 2 MPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MPLL_TMONCLKSEL				MPLL_RELATCHEN	MPLL_VCOVTSEL		MPLL_VCOCALSELB		MPLL_VCOCALSELA		MPLL_VCOCAL_EN	MPLL_VCOBAND			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	

MPLL_TMONCLKSEL MPLL Testmode Clock Selection (for DEBUG only)

MPLL_RELATCHEN MPLL Feedback Divider Relatch Function Enable (for DEBUG only)

MPLL_VCOVTSEL MPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

MPLL_VCOCALSELB MPLL VCOCAL Period B Selection (for DEBUG only)

MPLL_VCOCALSELA MPLL VCOCAL Period A Selection (for DEBUG only)

MPLL_VCOCAL_EN MPLL VCOCAL Function Enable.

- **0** Disable.
- **1** Enable. (default).

MPLL_VCOBAND MPLL VCO BAND Selection (for DEBUG only)

0x80060028h MCU (DSP PLL) Control Register 3 MPLL3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											MPLL_FBDIV					
Type											R/W					
Reset											011110					

MPLL_FBDIV MPLL Feedback Divider Ratio Control in analog part. divider ratio= MPLL_FBDIV[5:0] + 2.
Analog frequency = 13MHz * (MPLL_FBDIV+2).
*** MPLL_FBDIV_should range from 6'd26 to 6'd34 (364MHz ~ 468MHz)

0x80060030h USB PLL Control Register UPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	UPLL_VCOCALOK	UPLL_CP_P							UPLL_RST				UPLL_CM			
Type	R	R/W							R/W				R/W			
Reset	-	000							0				0000			

UPLL_VCOCALOK UPLL VCOBAND Calibration OK

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective when UPLL_VCOCAL_EN (in register UPLL2) is high.

UPLL_CP_P UPLL BANDWIDTH CONTROL (for DEBUG only)

UPLL_RST Reset Control of UPLL

- **0** Normal Operation
- **1** Reset the PLL

UPLL_CM UPLL Capacitor multiplier ratio

0x80060034h USB PLL Control Register 2 UPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	UPLL_TMONCLKSEL				UPLL_RELATCHEN	UPLL_VCOVTSEL		UPLL_VCOCALSELB		UPLL_VCOCALSELA		UPLL_VCOCAL_EN	UPLL_VCOBAND			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	

UPLL_TMONCLKSEL UPLL Testmode Clock Selection (for DEBUG only)

UPLL_RELATCHEN UPLL Feedback Divider Relatch Function Enable (for DEBUG only)

UPLL_VCOVTSEL UPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

UPLL_VCOCALSELB UPLL VCOCAL Period B Selection (for DEBUG only)

UPLL_VCOCALSELA UPLL VCOCAL Period A Selection (for DEBUG only)

UPLL_VCOCAL_EN UPLL VCOCAL Function Enable (for DEBUG only)

- **0** Disable.
- **1** Enable. (default).

UPLL_VCOBAND UPLL VCO BAND Selection (for DEBUG only)

0x80060038h Camera PLL Control Register CPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CPLL_VCOCALOK	CPLL_CP_P			CPLL_RST	CPLL_FBDIV						CPLL_CM				
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

CPLL is a PLL dedicated for output clock feeding to backend module. The PLL clock range from 13MHz to 208MHz with step of 3.25MHz. In view of analog block, it only provides clocks from 104MHz to 208MHz, controlled by CPLL_FBDIV.

As for the lower frequency range (13MHz ~ 104MHz), it is derived with aid of digital divider, controlled by CPLL_DIGDIV (Please see register “CPLL3” for details). For example, to derive 104MHz, software can make decision that analog block directly give 104MHz, or generate 208MHz then digitally divided by 2

CPLL_VCOCALOK CPLL VCOBAND Calibration OK

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective when CPLL_VCOCAL_EN (in register CPLL2) is high.

CPLL_CP_P CPLL BANDWIDTH CONTROL (for DEBUG only)

CPLL_RST Reset Control of CPLL

- **0** Normal Operation
- **1** Reset the PLL

CPLL_FBDIV CPLL Feedback Divider Ratio Control in analog part. Divider Ratio= CPLL_FBDIV[5:0] + 2. Analog frequency = 3.25MHz x (CPLL_FBDIV+2). CPLL_FBDIV_should range from 30 to 62.

CPLL_CM CPLL Capacitor multiplier ratio (for DEBUG only)

0x8006003Ch Camera PLL Control Register 2 CPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CPLL_TMONCLKSEL				CPLL_RELATCHEN	CPLL_VCOVTSEL		CPLL_VCOCALSELB		CPLL_VCOCALSELA		CPLL_VCOCAL_EN	CPLL_VCOBAND			CPLL_PWDB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

CPLL_TMONCLKSEL CPLL Testmode Clock Selection (for DEBUG only)

CPLL_RELATCHEN CPLL Feedback Divider Relatch Function Enable (for DEBUG only)

CPLL_VCOVTSEL CPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

CPLL_VCOCALSELB CPLL VCOCAL Period B Selection (for DEBUG only)

CPLL_VCOCALSELA CPLL VCOCAL Period A Selection (for DEBUG only)

CPLL_VCOCAL_EN CPLL VCOCAL Function Enable (for DEBUG only)

- **0** Disable.
- **1** Enable. (default).

CPLL_VCOBAND CPLL VCO BAND Selection (for DEBUG only)

CPLL_PWDB Control CPLL power-down (Active-Low)

- **0** Power-down
- **1** Power-on

0x80060048h Camera PLL Control Register 3 CPLL3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CPLL_DIGDIV			
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

CPLL_DIGDIV Camera PLL divider in digital part. It should range within {0, 1, 3, 7, 15}.

- **0000** Output camera pll's output directly, i.e. divided by 1.
- **0001** divided by 2.
- **0011** divided by 4.
- **0111** divided by 8.
- **1111** divided by 16.

0x80060040h TV PLL Control Register TPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TPLL_VCOCALOK	TPLL_CP_P			TPLL_RST								TPLL_CM			
Type	R	R/W	R/W	R/W	R/W								R/W	R/W	R/W	R/W
Reset	-	0	0	0	0								0	0	0	0

TPLL_VCOCALOK TPLL VCOBAND Calibration OK

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective when TPLL_VCOCAL_EN (in register TPLL2) is high.

TPLL_CP_P TPLL BANDWIDTH CONTROL (for DEBUG only)

TPLL_RST Reset Control of TPLL

- **0** Normal Operation
- **1** Reset the PLL

TPLL_CM TPLL Capacitor multiplier ratio (for DEBUG only)

0x80060044h TV PLL Control Register 2 — TPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TPLL_TMONCLKSEL				TPLL_RELATCHEN	TPLL_VCOVTSEL		TPLL_VCOCALSELB		TPLL_VCOCALSELA		TPLL_VCOCAL_EN	TPLL_VCOBAND			TPLL_PWDB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

TPLL_TMONCLKSEL TPLL Testmode Clock Selection (for DEBUG only)

TPLL_RELATCHEN TPLL Feedback Divider Relatch Function Enable (for DEBUG only)

TPLL_VCOVTSEL TPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

TPLL_VCOCALSELB TPLL VCOCAL Period B Selection (for DEBUG only)

TPLL_VCOCALSELA TPLL VCOCAL Period A Selection (for DEBUG only)

TPLL_VCOCAL_EN TPLL VCOCAL Function Enable (for DEBUG only)

- **0** Disable.
- **1** Enable. (default).

TPLL_VCOBAND TPLL VCO BAND Selection (for DEBUG only)

TPLL_PWDB Control TPLL power-down (Active-Low)

- **0** Power-down
- **1** Power-on

0x80060058h Memory Card PLL Control Register — MCPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MCPLL_VCOCALOK	MCPLL_CP_P			MCPLL_RST								MCPLL_CM			
Type	R	R/W	R/W	R/W	R/W								R/W	R/W	R/W	R/W
Reset	-	0	0	0	0								0	0	0	0

MCPLL is a PLL dedicated for output clock feeding to the memory card module. In view of analog block, it only provides 91MHz clock.

MCPLL_VCOCALOK MCPLL VCOBAND Calibration OK

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective whenM CPLL_VCOCAL_EN (in register MCPLL2) is high.

MCPLL_CP_P MCPLL BANDWIDTH CONTROL (for DEBUG only)

MCPLL_RST Reset Control of MCPLL

- **0** Normal Operation
- **1** Reset the PLL

MCPLL_CM MCPLL Capacitor multiplier ratio (for DEBUG only)

0x8006005Ch Memory Card PLL Control Register 2 MCPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MCPLL_TMONCLKSEL				MCPLL_RELATCHEN	MCPLL_VCOVTSEL		MCPLL_VCOCALSELB		MCPLL_VCOCALSELA		MCPLL_VCOCAL_EN	MCPLL_VCOBAND			MCPLL_PWDB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

MCPLL_TMONCLKSEL MCPLL Testmode Clock Selection (for DEBUG only)

MCPLL_VCOVTSEL MCPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

MCPLL_VCOCALSELB MCPLL VCOCAL Period B Selection (for DEBUG only)

MCPLL_VCOCALSELA MCPLL VCOCAL Period A Selection (for DEBUG only)

MCPLL_VCOCAL_EN MCPLL VCOCAL Function Enable (for DEBUG only)

- **0** Disable.
- **1** Enable. (default).

MCPLL_VCOBAND MCPLL VCO BAND Selection (for DEBUG only)

MCPLL_PWDB Control MCPLL power-down (Active-Low)

- **0** Power-down
- **1** Power-on

0x80060060h CEVAPLL Control Register CEVAPLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CEVAPLL_VCOCALOK	CEVAPLL_CP_P			CEVAPLL_RST	CEVAPLL_DIGDIV	CEVAPLL_FBDIV						CEVAPLL_CM			
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0

CEVAPLL is a PLL dedicated for output clock feeding to the CEVA DSP. The PLL clock range from 104MHz to 416MHz with step of 13MHz. In view of analog block, it only provides clocks from 208MHz to 416MHz, controlled by CEVAPLL_DIVCTRL. As for the lower frequency range (104MHz ~ 208MHz), it is derived with aid of digital divider, controlled by CEVAPLL_DIGDIV. For example, to derive 104MHz, software can make decision that analog block gives 208MHz, then digitally divided by 2; or analog block directly give 104MHz clock.

CEVAPLL_VCOCALOK CEVAPLL VCOBAND Calibration OK.

- **0** PLL auto calibration has not done.
- **1** PLL auto calibration is ok.

Note that this bit is only effective whenM CPLL_VCOCAL_EN (in register CEVAPLL2) is high.

CEVAPLL_CP_P CEVAPLL BANDWIDTH CONTROL (for DEBUG only)

CEVAPLL_DIGDIV CEVA PLL divider in digital part.

- **0** Output camera pll's output directly, i.e. divided by 1
- **1** divided by 2.

CEVAPLL_FBDIV CEVAPLL Feedback Divider Ratio Control in analog part. Divider Ratio= MPLL_FBDIV[5:0] + 2. Analog frequency = 13MHz x (CEVAPLL_FBDIV+2). CEVAPLL_FBDIV_should range from 14 to 30

CEVAPLL_RST Reset Control of CEVAPLL

- **0** Normal Operation
- **1** Reset the PLL

CEVAPLL_CM CEVAPLL Capacitor multiplier ratio (for DEBUG only)

0x80060064h CEVAPLL Control Register 2 CEVAPLL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CEVAPLL_TMONCLKSEL				CEVAPLL_RELATCHEN	CEVAPLL_VCOVTSEL		CEVAPLL_VCOCALSELB		CEVAPLL_VCOCALSELA		CEVAPLL_VCOCAL_EN	CEVAPLL_VCOBAND			CEVAPLL_PWDB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

CEVAPLL_TMONCLKSEL CEVAPLL Testmode Clock Selection (for DEBUG only)

CEVAPLL_VCOVTSEL CEVAPLL VCOCAL Slicer Voltage Selection (for DEBUG only)

CEVAPLL_VCOCALSELB CEVAPLL VCOCAL Period B Selection (for DEBUG only)

CEVAPLL_VCOCALSELA CEVAPLL VCOCAL Period A Selection (for DEBUG only)

CEVAPLL_VCOCAL_EN CEVAPLL VCOCAL Function Enable (for DEBUG only)

- **0** Disable.
- **1** Enable. (default).

CEVAPLL_VCOBAND CEVAPLL VCO BAND Selection (for DEBUG only)

CEVAPLL_PWDB Control CEVAPLL power-down (Active-Low)

- **0** Power-down
- **1** Power-on

0x80060050h PLL Bias Control PLL_BIAS

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RG_PLL_VLDO_CALI				RG_CALI_BIAS						CLKSQ_DIV2_SEL		RG_PLL_TESTMODE			
Type	R/W				R/W						R/E		R/W			
Reset	0000				0000						1		0000			

RG_PLL_VLDO_CALI PLL LDO Calibration. (for DEBUG only)

RG_PLLBIAS_CAI PLL LDO Bias Calibration. (for DEBUG only)

CLKSQ_DIV2_SEL Indicator of PLL input frequency from CLKSQ.

- **0** Send CLKSQ divided by 1 clock to PLL input
- **1** Send CLKSQ divided by 2 clock to PLL input

RG_PLL_TESTMDOE PLL testmode (for DEBUG only)

0x8006004C PLL Reserved Control Register 0 PLL_RES_CON 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RG_PLL_RESB				RG_PLL_RESA			
Type									R/W				R/W			
Reset									0000				0000			

RG_PLL_RESA[0] MIPI Macro reference clock source disable.

- **0** Send the 26MHz clock to MIPI Macro as its reference clock.
- **1** Disable the 26MHz clock to MIPI Macro.

RG_PLL_RESA[3:1] Reserved A-set. (for DEBUG only)

RG_PLL_RESB Reserved B-set (for DEBUG only)

0x80060070h IDN Control IDN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													UPLL_IDN	DPLL_IDN	MPLL_IDN	CLKSQ_IDN
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

The register is for debug usage. In normal function, it can power-on CLKSQ or corresponding PLL without regards to SRCLKENA (ie. sleep-mode indication) but still need to set PDN_CON at first. In ACD test mode, it can power-on corresponding PLL directly (CLKSQ excluded), but still need to set PDN_CON at first.

0x8006007Ch XOSC32 Analog Circuit Control XOSC32_CON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PWDB	CALI				
Type											R/W	R/W				
Reset											0	00101				

The register is for debug usage. Only when MT6516 is in STAMP mode, these register fields will take effect.

PWDB Power down XOSC32 circuit (Active-low). It is only valid when MT6516 is in STAMP mode. When MT6516 is in normal mode, XOSC32 is always ON.

- **0** Power down.
- **1** Power on.

CALI[4:0] gm value of XOSC32 circuit. It is only valid when MT6516 is in STAMP mode. When MT6516 is in normal mode, XOSC32's gm value is tied to '00101'.

6.3 Pulse-Width Modulation Outputs

6.3.1 General Description

Six generic pulse-width modulators are implemented to generate pulse sequences with programmable frequency and duration for LCD backlight, charging or other purpose. Before enabling PWM, the pulse sequences must be prepared either in the memory or registers. Then PWM, as shown in Fig. 36, will read the pulse sequences to generate random waveform to meet all kinds of applications.

 © 2009 MediaTek Inc.

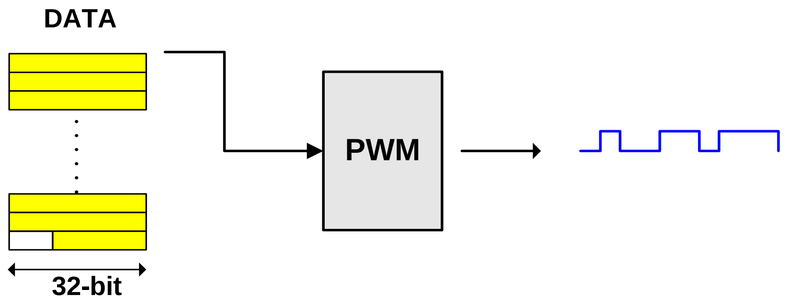


Fig. 36 The generation procedure of PWM.

There are two basic operational modes about PWM, which is set by PWM_MODE. In periodical mode, all pulse sequence with be repeatedly generated by the number of WAVE_NUM[15:0]. If WAVE_NUM is 0 which means infinite, the waveform generation could be stopped by PWM_EN. As for the pulse sequence data source in the periodical mode, if the data are less than or equal to 64 bits, they can be directly set in SEND_DATA0[31:0] and SEND_DATA1[31:0] and SRCSEL=0 to reduce memory bandwidth. STOP_BITPOS[5:0] is used to indicate the stop bit position in the total 64-bits data. For example, if STOP_BITPOS is 0, only SEND_DATA0[0] will be generated, and so on until SEND_DATA1[31]. If SRCSEL=1 which means memory mode, the pulse sequence data are put in memory with address set by BUF0_BASE_ADDR and the length is BUF0_SIZE. STOP_BITPOS[4:0] is to indicate the stop bit position in the last 32-bits data. The format of pulse sequences that stored in periodical mode is as shown in Fig. 37.

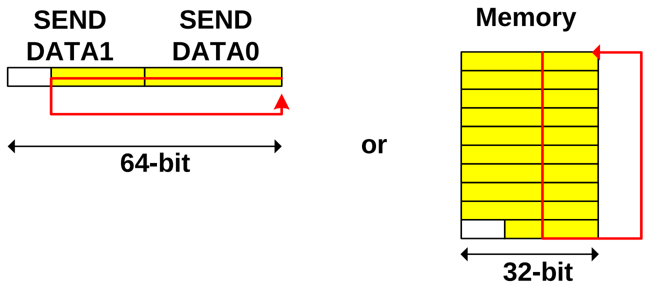


Fig. 37 The pulse sequence in periodical mode.

On the other hand, the pulse sequence is stored in dual memory buffers in random mode. The format of pulse sequences that stored in the memory is as shown in Fig. 38. Valid bit is used to indicate data are ready in the respective memory buffer. The PWM generation will clear this bit after all data in that buffer are fetched. The memory buffers are set by address BUF0_BASE_ADDR and BUF0_SIZE for memory buffer0 and BUF1_BASE_ADDR and BUF1_SIZE for memory buffer1. The program should prepare the pulse sequence and set the valid to 1 in time before all data in the other memory buffer are fetched or the HW will issue UNDERFLOW interrupt to inform pulse generation will be stopped because of no valid data.

© 2009 MediaTek Inc.

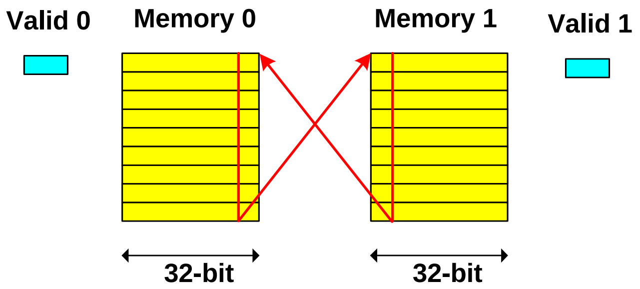


Fig. 38 The pulse sequence in random mode

PWM always reference bus block clock (52MHz) as base, and CLKDIV[2:0] and CLKSEL can decide the sample rate of each PWM. When system is in the sleep mode, block clock will be disabled and only OLD_PWM_MODE with CLKSEL=1 (32 KHz) is supported. Only PWM1, PWM2 and PWM3 support OLD_PWM_MODE. For each sample output, the duration is decided by HDURATION[15:0] when output is high and LDURATION[15:0] when output is low. If the pulse sequence is repeated which is specified by WAVE_NUM[15:0], a special output could be set by GUARD_VALUE and GUARD_DURATION[15:0] between these pulse sequence. The PWM output will be the value specified by IDLE_VALUE when PWM is not enabled or the pulse sequence is finished.

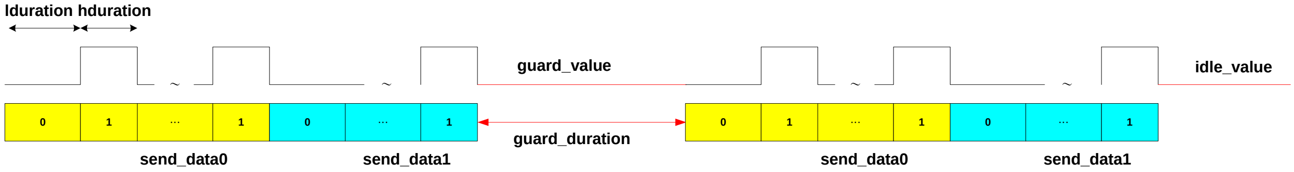


Fig. 39 The pulse sequence output pattern

In order to provide precise timing relation between different PWM outputs, we provide PWM_SEQ_MODE. In this mode, the starting position of waveform outputs of PWM3, PWM4, PWM5 and PWM6 will follow the previous one by the delay values PWM4_DEALY_DURATION[15:0], PWM5_DEALY_DURATION[15:0] and PWM6_DEALY_DURATION[15:0]. Also the clock scale of each delay can be specified by PWM4_DELAY_CLKSEL, PWM5_DELAY_CLKSEL and PWM6_DELAY_CLKSEL.

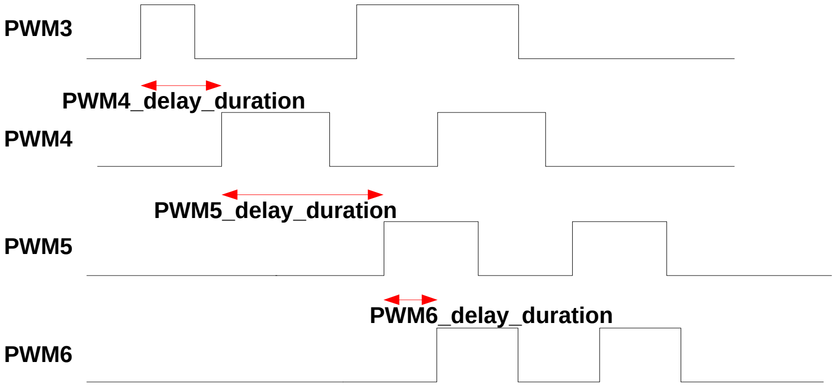


Fig. 40 The sequential output mode

 © 2009 MediaTek Inc.

Also PWM1, PWM2 and PWM3 support original PWM output mode. The output waveform is specified by DATA_WIDTH[12:0] and THRESH[12:0]. The output waveform is shown in Fig. 41.

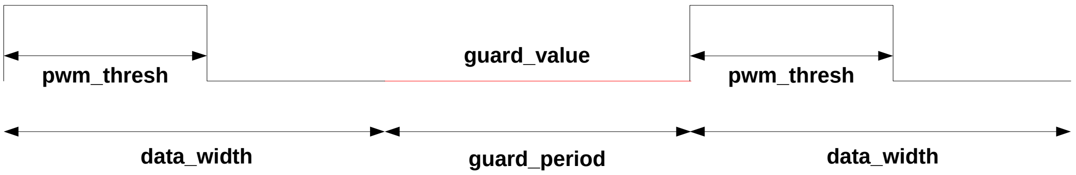


Fig. 41 The old PWM mode

For hardware and system consideration, CLKSRC might be slightly different in different situations. The following table is to summary all possible situations.

PWM_OLD_MODE	PWM_CLKSEL	CLKSRC
1	0	block clock
	1	32 KHz
0	0	block clock
	1	block clock / 1625

6.3.2 Register Table

Register Address	Register Function	Acronym
PWM + 0000h	PWM enable register	**PWM_ENABLE**
PWM + 0004h	PWM4 delay duration register	**PWM4_DELAY**
PWM + 0008h	PWM5 delay duration register	**PWM5_DELAY**
PWM + 000Ch	PWM6 delay duration register	**PWM6_DELAY**
PWM + 0010h	PWM1 control register	**PWM1_CON**
PWM + 0014h	PWM1 high duration register	**PWM1_HDURATION**
PWM + 0018h	PWM1 low duration register	**PWM1_LDURATION**
PWM + 001Ch	PWM1 guard duration register	**PWM1_GDURATION**
PWM + 0020h	PWM1 buffer0 base address register	**PWM1_BUF0_BASE_ADDR**
PWM + 0024h	PWM1 buffer0 size register	**PWM1_BUF0_SIZE**
PWM + 0028h	PWM1 buffer1 base address register	**PWM1_BUF1_BASE_ADDR**
PWM + 002Ch	PWM1 buffer1 size register	**PWM1_BUF1_SIZE**
PWM + 0030h	PWM1 send data0 register	**PWM1_SEND_DATA0**
PWM + 0034h	PWM1 send data1 register	**PWM1_SEND_DATA1**
PWM + 0038h	PWM1 wave number register	**PWM1_WAVE_NUM**

PWM + 003Ch	PWM1 data width	**PWM1_DATA_WIDTH**
PWM + 0040h	PWM1 threshold register	**PWM1_THRESH**
PWM + 0044h	PWM1 send waveform number register	**PWM1_SEND_WAVENUM**
PWM + 0048h	PWM1 valid register	**PWM1_VALID**
PWM + 0050h	PWM2 control register	**PWM2_CON**
PWM + 0054h	PWM2 high duration register	**PWM2_HDURATION**
PWM + 0058h	PWM2 low duration register	**PWM2_LDURATION**
PWM + 005Ch	PWM2 guard duration register	**PWM2_GDURATION**
PWM + 0060h	PWM2 buffer0 base address register	**PWM2_BUF0_BASE_ADDR**
PWM + 0064h	PWM2 buffer0 size register	**PWM2_BUF0_SIZE**
PWM + 0068h	PWM2 buffer1 base address register	**PWM2_BUF1_BASE_ADDR**
PWM + 006Ch	PWM2 buffer1 size register	**PWM2_BUF1_SIZE**
PWM + 0070h	PWM2 send data0 register	**PWM2_SEND_DATA0**
PWM + 0074h	PWM2 send data1 register	**PWM2_SEND_DATA1**
PWM + 0078h	PWM2 wave number register	**PWM2_WAVE_NUM**
PWM + 007Ch	PWM2 data width	**PWM2_DATA_WIDTH**
PWM + 0080h	PWM2 threshold register	**PWM2_THRESH**
PWM + 0084h	PWM2 send waveform number register	**PWM2_SEND_WAVENUM**
PWM + 0088h	PWM2 valid register	**PWM2_VALID**
PWM + 0090h	PWM3 control register	**PWM3_CON**
PWM + 0094h	PWM3 high duration register	**PWM3_HDURATION**
PWM + 0098h	PWM3 low duration register	**PWM3_LDURATION**
PWM + 009Ch	PWM3 guard duration register	**PWM3_GDURATION**
PWM + 00A0h	PWM3 buffer0 base address register	**PWM3_BUF0_BASE_ADDR**
PWM + 00A4h	PWM3 buffer0 size register	**PWM3_BUF0_SIZE**
PWM + 00A8h	PWM3 buffer1 base address register	**PWM3_BUF1_BASE_ADDR**
PWM + 00ACh	PWM3 buffer1 size register	**PWM3_BUF1_SIZE**
PWM + 00B0h	PWM3 send data0 register	**PWM3_SEND_DATA0**
PWM + 00B4h	PWM3 send data1 register	**PWM3_SEND_DATA1**
PWM + 00B8h	PWM3 wave number register	**PWM3_WAVE_NUM**
PWM + 00BCh	PWM3 data width	**PWM3_DATA_WIDTH**
PWM + 00C0h	PWM3 threshold register	**PWM3_THRESH**
PWM + 00C4h	PWM3 send waveform number register	**PWM3_SEND_WAVENUM**
PWM + 00C8h	PWM3 valid register	**PWM3_VALID**
PWM + 00D0h	PWM4 control register	**PWM4_CON**
PWM + 00D4h	PWM4 high duration register	**PWM4_HDURATION**

PWM + 00D8h	PWM4 low duration register	**PWM4_LDURATION**
PWM + 00DCh	PWM4 guard duration register	**PWM4_GDURATION**
PWM + 00E0h	PWM4 buffer0 base address register	**PWM4_BUF0_BASE_ADDR**
PWM + 00E4h	PWM4 buffer0 size register	**PWM4_BUF0_SIZE**
PWM + 00E8h	PWM4 buffer1 base address register	**PWM4_BUF1_BASE_ADDR**
PWM + 00ECh	PWM4 buffer1 size register	**PWM4_BUF1_SIZE**
PWM + 00F0h	PWM4 send data0 register	**PWM4_SEND_DATA0**
PWM + 00F4h	PWM4 send data1 register	**PWM4_SEND_DATA1**
PWM + 00F8h	PWM4 wave number register	**PWM4_WAVE_NUM**
PWM + 00FCh	PWM4 send waveform number register	**PWM4_SEND_WAVENUM**
PWM + 0100h	PWM4 valid register	**PWM4_VALID**
PWM + 0110h	PWM5 control register	**PWM5_CON**
PWM + 0114h	PWM5 high duration register	**PWM5_HDURATION**
PWM + 0118h	PWM5 low duration register	**PWM5_LDURATION**
PWM + 011Ch	PWM5 guard duration register	**PWM5_GDURATION**
PWM + 0120h	PWM5 buffer0 base address register	**PWM5_BUF0_BASE_ADDR**
PWM + 0124h	PWM5 buffer0 size register	**PWM5_BUF0_SIZE**
PWM + 0128h	PWM5 buffer1 base address register	**PWM5_BUF1_BASE_ADDR**
PWM + 012Ch	PWM5 buffer1 size register	**PWM5_BUF1_SIZE**
PWM + 0130h	PWM5 send data0 register	**PWM5_SEND_DATA0**
PWM + 0134h	PWM5 send data1 register	**PWM5_SEND_DATA1**
PWM + 0138h	PWM5 wave number register	**PWM5_WAVE_NUM**
PWM + 013Ch	PWM5 send waveform number register	**PWM5_SEND_WAVENUM**
PWM + 0140h	PWM5 valid register	**PWM5_VALID**
PWM + 0150h	PWM6 control register	**PWM6_CON**
PWM + 0154h	PWM6 high duration register	**PWM6_HDURATION**
PWM + 0158h	PWM6 low duration register	**PWM6_LDURATION**
PWM + 015Ch	PWM6 guard duration register	**PWM6_GDURATION**
PWM + 0160h	PWM6 buffer0 base address register	**PWM6_BUF0_BASE_ADDR**
PWM + 0164h	PWM6 buffer0 size register	**PWM6_BUF0_SIZE**
PWM + 0168h	PWM6 buffer1 base address register	**PWM6_BUF1_BASE_ADDR**
PWM + 016Ch	PWM6 buffer1 size register	**PWM6_BUF1_SIZE**
PWM + 0170h	PWM6 send data0 register	**PWM6_SEND_DATA0**
PWM + 0174h	PWM6 send data1 register	**PWM6_SEND_DATA1**
PWM + 0178h	PWM6 wave number register	**PWM6_WAVE_NUM**
PWM + 017Ch	PWM6 send waveform number register	**PWM6_SEND_WAVENUM**
PWM + 0180h	PWM6 valid register	**PWM6_VALID**

 © 2009 MediaTek Inc.

PWM + 0190h	PWM interrupt enable register	**PWM_INT_ENABLE**
PWM + 0194h	PWM interrupt status register	**PWM_INT_STATUS**
PWM + 0198h	PWM interrupt acknowledge register	**PWM_INT_ACK**
PWM + 01A0h	PWM0 control register	**PWM0_CON**
PWM + 01A4h	PWM0 high duration register	**PWM0_HDURATION**
PWM + 01A8h	PWM0 low duration register	**PWM0_LDURATION**
PWM + 01Ach	PWM0 guard duration register	**PWM0_GDURATION**
PWM + 01B0h	PWM0 buffer0 base address register	**PWM0_BUF0_BASE_ADDR**
PWM + 01B4h	PWM0 buffer0 size register	**PWM0_BUF0_SIZE**
PWM + 01B8h	PWM0 buffer1 base address register	**PWM0_BUF1_BASE_ADDR**
PWM + 01BCh	PWM0 buffer1 size register	**PWM0_BUF1_SIZE**
PWM + 01C0h	PWM0 send data0 register	**PWM0_SEND_DATA0**
PWM + 01C4h	PWM0 send data1 register	**PWM0_SEND_DATA1**
PWM + 01C8h	PWM0 wave number register	**PWM0_WAVE_NUM**
PWM + 01CCh	PWM0 data width	**PWM0_DATA_WIDTH**
PWM + 01D0h	PWM0 threshold register	**PWM0_THRESH**
PWM + 01D4h	PWM0 send waveform number register	**PWM0_SEND_WAVENUM**
PWM + 01D8h	PWM0 valid register	**PWM0_VALID**

Table 156 ***PWM Registers***

6.3.3 Register Definitions

PWM+0000h PWM Enable register PWM_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								PWM_TEST_SEL	PWM_SEQ_MODE	PWM6_EN	PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN
Type								RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								0	0	0	0	0	0	0	0	0

PWM0_EN Set to 1 to enable PWM0
PWM1_EN Set to 1 to enable PWM1
PWM2_EN Set to 1 to enable PWM2
PWM3_EN Set to 1 to enable PWM3
PWM4_EN Set to 1 to enable PWM4
PWM5_EN Set to 1 to enable PWM5
PWM6_EN Set to 1 to enable PWM6

Note : When turning off PWM unit (PWM_EN = 1→0), PWM needs some clock periods (32kHz or 52MHz clock) to shut down.

PWM_SEQ_MODE Set to 1 to enable PWM3, PWM4, PWM5 and PWM6 sequential delay mode. In this mode, PWM3 starts first and then after PWM4_DELAY_TIME, PWM4 will start. After PWM4 starts, PWM5 will start after PWM5_DELAY_TIME and so on for PWM6.

Note: The output of PWM_SEQ_MODE is started after PWM3 is enabled. And PWM_SEQ_MODE should be set before PWM4, PWM5 and PWM6 are enabled or at the same time. Also this mode doesn't work when PWM3 is set at OLD_PWM_MODE and CLKSEL=1.

PWM_TEST_SEL Set to 1 to enable the switch of the PWM output signal between pwm unit1, pwm unit2 and pwm unit5, pwm unit6. The default (0) behavior is to select the output of pwm unit5 and pwm unit6. If set to 1, the output of pwm unit1 will be selected instead of pwm unit5, and the output of pwm unit2 will be selected instead of pwm unit6.

PWM+0004h PWM4 Delay Duration register PWM4_DELAY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**DELAY_CLKSEL**
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PWM4_DELAY_DURATION[15:0]**															
Type	R/W															
Reset	0															

PWM4_DELAY_DURATION The time difference between PWM3 and PWM4.

DELAY_CLKSEL The clock unit of PWM4_DELAY_DURATION.

- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

PWM+0008h PWM5 Delay Duration register PWM5_DELAY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**DELAY_CLKSEL**
Type																R/W
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PWM5_DELAY_DURATION[15:0]**															
Type	R/W															
Reset	0															

PWM5_DELAY_DURATION The time difference between PWM4 and PWM5.

DELAY_CLKSEL The clock unit of PWM5_DELAY_DURATION.

- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

PWM+000Ch PWM6 Delay Duration register PWM6_DELAY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**DELAY_CLKSEL**

Type															R/W
Reset															0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM6_DELAY_DURATION[15:0]															
Type	R/W															
Reset	0															

PWM6_DELAY_DURATION The time difference between PWM5 and PWM6.

DELAY_CLKSEL The clock unit of PWM6_DELAY_DURATION.

0 CLK=CLKSRC

1 CLK=CLKSRC/1625

PWM+0010h PWM1 Control register PWM1_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM1 clock scale.

000 CLK Hz

001 CLK/2 Hz

010 CLK/4 Hz

011 CLK/8 Hz

100 CLK/16 Hz

101 CLK/32 Hz

110 CLK/64 Hz

111 CLK/128 Hz

CLKSEL Select PWM1 clock

0 CLK=CLKSRC

1 CLK=CLKSRC/1625

SRCSEL Select PWM1 data source

0 FIFO mode

1 Memory mode

MODE Select Random Generator mode

0 Periodical PWM mode.

1 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM1 output value when idle state.

GUARD_VALUE PWM1 output value when guard time.

STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.

OLD_PWM_MODE Use old PWM mode

0 New PWM mode
1 Old PWM mode

Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source could work in the system sleep-mode.

PWM+0014h PWM1 High Duration register PWM1_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION[15:0]															
Type	R/W															
Reset	1															

HDURATION PWM1 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+0018h PWM1 Low Duration register PWM1_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION[15:0]															
Type	R/W															
Reset	1															

LDURATION PWM1 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+001Ch PWM1 Guard Duration register PWM1_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION[15:0]															
Type	R/W															
Reset	0															

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

PWM+0020h PWM1 Buffer0 Base Address register PWM1_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF0_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM1's waveform data.

PWM+0024h PWM1 Buffer0 Size register PWM1_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM1 should generate. If it equals to N, need to program N-1 in this register.

Note: The size is in unit of 32-bit data.

PWM+0028h PWM1 Buffer1 Base Address register PWM1_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF1_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM1's waveform data.

Note: The memory buffer1 is useless in periodical mode.

PWM+002Ch PWM1 Buffer1 Size register PWM1_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_SIZE[15:0]															

© 2009 MediaTek Inc.

Type	R/W
Reset	0

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM1 should generate. If it equals to N, need to program N-1 in this register.

PWM+0030h PWM1 Send Data0 register PWM1_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA0 [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA0[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA0 PWM1 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0034h PWM1 Send Data1 register PWM1_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA1[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA1 PWM1 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0038h PWM1 Wave Number register PWM1_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAVE_NUM[15:0]**															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM1 will generate from the pulse data repeatedly.

Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+003Ch PWM1 Data Width register PWM1_DATA_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DATA_WIDTH[12:0]												
Type				R/W												
Reset				0												

DATA_WIDTH The PWM1 pulse data width in the old PWM mode.

PWM+0040h PWM1 Thresh register PWM1_THRESH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				THRESH[12:0]												
Type				R/W												
Reset				0												

THRESH The PWM1 pulse data high/low switching threshold in the old PWM mode.

PWM+0044h PWM1 Send Wave Number register PWM1_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM1 has already generated from the specified data source in the periodical mode.

PWM+0048h PWM1 Valid register PWM1_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+0050h PWM2 Control register PWM2_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM2 clock scale.
- **000** CLK Hz
- **001** CLK/2 Hz
- **010** CLK/4 Hz
- **011** CLK/8 Hz
- **100** CLK/16 Hz
- **101** CLK/32 Hz
- **110** CLK/64 Hz
- **111** CLK/128 Hz

CLKSEL Select PWM1 clock
- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

SRCSEL Select PWM2 data source
- **0** FIFO mode
- **1** Memory mode

MODE Select Random Generator mode
- 0 Periodical PWM mode.
- 1 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM2 output value when idle state.
GUARD_VALUE PWM2 output value when guard time.
STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.
OLD_PWM_MODE Use old PWM mode
- 0 New PWM mode
- 1 Old PWM mode

Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source could work in the system sleep-mode.

PWM+0054h PWM2 High Duration register PWM2_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION[15:0]															
Type	R/W															
Reset	1															

HDURATION PWM2 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+0058h PWM2 Low Duration register PWM2_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION[15:0]															
Type	R/W															
Reset	1															

LDURATION PWM2 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+005Ch PWM2 Guard Duration register PWM2_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION[15:0]															
Type	R/W															
Reset	0															

GUARD_DURATION It’s the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

© 2009 MediaTek Inc.

PWM+0060h PWM2 Buffer0 Base Address register PWM2_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF0_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM2's waveform data.

PWM+0064h PWM2 Buffer0 Size register PWM2_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM2 should generate. If it equals to N, need to program N-1 in this register.

PWM+0068h PWM2 Buffer1 Base Address register PWM2_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF1_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM2's waveform data.

Note: The memory buffer1 is useless in periodical mode.

PWM+006Ch PWM2 Buffer1 Size register PWM2_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_SIZE[15:0]															
Type	R/W															

© 2009 MediaTek Inc.

Reset	0

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM2 should generate. If it equals to N, need to program N-1 in this register.

PWM+0070h PWM2 Send Data0 register PWM2_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0 [31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA0[15:0]															
Type	R/W															
Reset	0															

SEND_DATA0 PWM2 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0074h PWM2 Send Data1 register PWM2_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	R/W															
Reset	0															

SEND_DATA1 PWM2 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0078h PWM2 Wave Number register PWM2_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM[15:0]															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM2 will generate from the pulse data repeatedly.

Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+007Ch PWM2 Data Width register PWM2_DATA_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DATA_WIDTH[12:0]												
Type				R/W												
Reset				0												

DATA_WIDTH The PWM2 pulse data width in the old PWM mode.

PWM+0080h PWM2 Thresh register PWM2_THRESH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				THRESH[12:0]												
Type				R/W												
Reset				0												

THRESH The PWM2 pulse data high/low switching threshold in the old PWM mode.

PWM+0084h PWM2 Send Wave Number register PWM2_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM2 has already generated from the specified data source in the periodical mode.

PWM+0088h PWM2 Valid register PWM2_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

 © 2009 MediaTek Inc.

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+0090h PWM3 Control register PWM3_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM3 clock scale.
- **000** CLK Hz
- **001** CLK/2 Hz
- **010** CLK/4 Hz
- **011** CLK/8 Hz
- **100** CLK/16 Hz
- **101** CLK/32 Hz
- **110** CLK/64 Hz
- **111** CLK/128 Hz

CLKSEL Select PWM1 clock
- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

SRCSEL Select PWM3 data source
- **0** FIFO mode
- **1** Memory mode

MODE Select Random Generator mode
- 0 Periodical PWM mode.
- 1 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM3 output value when idle state.
GUARD_VALUE PWM3 output value when guard time.
STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.
OLD_PWM_MODE Use old PWM mode
- 0 New PWM mode
- 1 Old PWM mode

Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source could work in the system sleep-mode.

PWM+0094h PWM3 High Duration register PWM3_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION[15:0]															
Type	R/W															
Reset	1															

HDURATION PWM3 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+0098h PWM3 Low Duration register PWM3_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION[15:0]															
Type	R/W															
Reset	1															

LDURATION PWM3 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+009Ch PWM3 Guard Duration register PWM3_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION[15:0]															
Type	R/W															
Reset	0															

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

© 2009 MediaTek Inc.

PWM+00A0h PWM3 Buffer0 Base Address register PWM3_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF0_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM3's waveform data.

PWM+00A4h PWM3 Buffer0 Size register PWM3_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM3 should generate. If it equals to N, need to program N-1 in this register.

PWM+00A8h PWM3 Buffer1 Base Address register PWM3_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF1_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM3's waveform data.

Note: The memory buffer1 is useless in periodical mode.

PWM+00ACh PWM3 Buffer1 Size register PWM3_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_SIZE[15:0]															
Type	R/W															

 © 2009 MediaTek Inc.

Reset	0

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM3 should generate. If it equals to N, need to program N-1 in this register.

PWM+00B0h PWM3 Send Data0 register PWM3_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0 [31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA0[15:0]															
Type	R/W															
Reset	0															

SEND_DATA0 PWM3 local buffer0 of pulse sequence data to be generated.
Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+00B4h PWM3 Send Data1 register PWM3_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	R/W															
Reset	0															

SEND_DATA1 PWM3 local buffer0 of pulse sequence data to be generated.
Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+00B8h PWM3 Wave Number register PWM3_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM[15:0]															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM3 will generate from the pulse data repeatedly.
Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+00BCh PWM3 Data Width register PWM3_DATA_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DATA_WIDTH[12:0]												
Type				R/W												
Reset				0												

DATA_WIDTH The PWM3 pulse data width in the old PWM mode.

PWM+00C0h PWM3 Thresh register PWM3_THRESH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				THRESH[12:0]												
Type				R/W												
Reset				0												

THRESH The PWM3 pulse data high/low switching threshold in the old PWM mode.

PWM+00C4h PWM3 Send Wave Number register PWM3_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM3 has already generated from the specified data source in the periodical mode.

PWM+00C8h PWM3 Valid register PWM3_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

© 2009 MediaTek Inc.

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.

BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.

BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.

BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.

Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+00D0h PWM4 Control register PWM4_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM4 clock scale.

- **000** CLK Hz
- **001** CLK/2 Hz
- **010** CLK/4 Hz
- **011** CLK/8 Hz
- **100** CLK/16 Hz
- **101** CLK/32 Hz
- **110** CLK/64 Hz
- **111** CLK/128 Hz

CLKSEL Select PWM1 clock

- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

SRCSEL Select PWM4 data source

- **0** FIFO mode
- **1** Memory mode

MODE Select Random Generator mode

- 0 Periodical PWM mode.
- 1 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM4 output value when idle state.

GUARD_VALUE PWM4 output value when guard time.

STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.

PWM+00D4h PWM4 High Duration register PWM4_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**HDURATION[15:0]**															
Type	R/W															
Reset	1															

HDURATION PWM4 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+00D8h PWM4 Low Duration register PWM4_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**LDURATION[15:0]**															
Type	R/W															
Reset	1															

LDURATION PWM4 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+00DCh PWM4 Guard Duration register PWM4_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**GUARD_DURATION[15:0]**															
Type	R/W															
Reset	0															

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

PWM+00E0h PWM4 Buffer0 Base Address register PWM4_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF0_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**BUF0_BS_ADDR[15:0]**															
Type	R/W															

Reset	0

BUF0_BS_ADDR The base address of memory buffer0 for PWM4's waveform data.

PWM+00E4h PWM4 Buffer0 Size register PWM4_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM4 should generate. If it equals to N, need to program N-1 in this register.

PWM+00E8h PWM4 Buffer1 Base Address register PWM4_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF1_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM4's waveform data.

Note: The memory buffer1 is useless in periodical mode.

PWM+00ECh PWM4 Buffer1 Size register PWM4_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM4 should generate. If it equals to N, need to program N-1 in this register.

PWM+00F0h PWM4 Send Data0 register PWM4_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0 [31:16]															
Type	R/W															

Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA0[15:0]															
Type	R/W															
Reset	0															

SEND_DATA0 PWM4 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+00F4h PWM4 Send Data1 register PWM4_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	R/W															
Reset	0															

SEND_DATA1 PWM4 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+00F8h PWM4 Wave Number register PWM4_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM[15:0]															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM4 will generate from the pulse data repeatedly.

Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+00FCh PWM4 Send Wave Number register PWM4_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

 © 2009 MediaTek Inc.

SEND_WAVENUM The number by which PWM4 has already generated from the specified data source in the periodical mode.

PWM+0100h PWM4 Valid register PWM4_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+0110h PWM5 Control register PWM5_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM5 clock scale.
- **000** CLK Hz
- **001** CLK/2 Hz
- **010** CLK/4 Hz
- **011** CLK/8 Hz
- **100** CLK/16 Hz
- **101** CLK/32 Hz
- **110** CLK/64 Hz
- **111** CLK/128 Hz

CLKSEL Select PWM1 clock
- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

SRCSEL Select PWM5 data source
- **0** FIFO mode
- **2** Memory mode

MODE Select Random Generator mode

2 Periodical PWM mode.
3 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM5 output value when idle state.

GUARD_VALUE PWM5 output value when guard time.

STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.

PWM+0114h PWM5 High Duration register PWM5_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION[15:0]															
Type	R/W															
Reset	1															

HDURATION PWM5 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+0118h PWM5 Low Duration register PWM5_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION[15:0]															
Type	R/W															
Reset	1															

LDURATION PWM5 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+011Ch PWM5 Guard Duration register PWM5_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION[15:0]															
Type	R/W															
Reset	0															

 © 2009 MediaTek Inc.

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.
Note: If this duration is 0, it means no guarding interval.

PWM+0120h PWM5 Buffer0 Base Address register PWM5_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF0_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF0_BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM5's waveform data.

PWM+0124h PWM5 Buffer0 Size register PWM5_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF0_SIZE[15:0]**															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM5 should generate. If it equals to N, need to program N-1 in this register.

PWM+0128h PWM5 Buffer1 Base Address register PWM5_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF1_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF1_BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM5's waveform data.
Note: The memory buffer1 is useless in periodical mode.

PWM+012Ch PWM5 Buffer1 Size register PWM5_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF1_SIZE[15:0]**															
Type	R/W															
Reset	0															

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM5 should generate. If it equals to N, need to program N-1 in this register.

PWM+0130h PWM5 Send Data0 register PWM5_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA0 [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA0[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA0 PWM5 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0134h PWM5 Send Data1 register PWM5_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA1[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA1 PWM5 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0138h PWM5 Wave Number register PWM5_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAVE_NUM[15:0]**															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM5 will generate from the pulse data repeatedly.

Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+013Ch PWM5 Send Wave Number register PWM5_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM5 has already generated from the specified data source in the periodical mode.

PWM+0140h PWM5 Valid register PWM5_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+0150h PWM6 Control register PWM6_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM6 clock scale.

000 CLK Hz
001 CLK/2 Hz
010 CLK/4 Hz

011 CLK/8 Hz
100 CLK/16 Hz
101 CLK/32 Hz
110 CLK/64 Hz
111 CLK/128 Hz

CLKSEL Select PWM1 clock
0 CLK=CLKSRC
1 CLK=CLKSRC/1625

SRCSEL Select PWM6 data source
0 FIFO mode
3 Memory mode

MODE Select Random Generator mode
4 Periodical PWM mode.
5 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM6 output value when idle state.

GUARD_VALUE PWM6 output value when guard time.

STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.

PWM+0154h PWM6 High Duration register PWM6_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**HDURATION[15:0]**															
Type	R/W															
Reset	1															

HDURATION PWM6 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+0158h PWM6 Low Duration register PWM6_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LDURATION[15:0]**															
Type	R/W															
Reset	1															

LDURATION PWM6 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+015Ch PWM6 Guard Duration register PWM6_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GUARD_DURATION[15:0]**															
Type	R/W															
Reset	0															

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

PWM+0160h PWM6 Buffer0 Base Address register PWM6_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF0_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF0_BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM6's waveform data.

PWM+0164h PWM6 Buffer0 Size register PWM6_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF0_SIZE[15:0]**															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM6 should generate. If it equals to N, need to program N-1 in this register.

PWM+0168h PWM6 Buffer1 Base Address register PWM6_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF1_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF1_BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM6's waveform data.
Note: The memory buffer1 is useless in periodical mode.

PWM+016Ch PWM6 Buffer1 Size register PWM6_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF1_SIZE[15:0]**															
Type	R/W															
Reset	0															

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM6 should generate. If it equals to N, need to program N-1 in this register.

PWM+0170h PWM6 Send Data0 register PWM6_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA0 [31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA0[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA0 PWM6 local buffer0 of pulse sequence data to be generated.
Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0174h PWM6 Send Data1 register PWM6_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA1[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA1 PWM6 local buffer0 of pulse sequence data to be generated.
Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+0178h PWM6 Wave Number register PWM6_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM[15:0]															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM6 will generate from the pulse data repeatedly.
Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+017Ch PWM6 Send Wave Number register PWM6_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM[15:0]															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM6 has already generated from the specified data source in the periodical mode.

PWM+0180h PWM6 Valid register PWM6_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													BUF1_VALID_WEN	BUF1_VALID	BUF0_VALID_WEN	BUF0_VALID
Type													W	R/W	W	R/W
Reset													0	0	0	0

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

PWM+0190h PWM Interrupt Enable register PWM_INT_ENABLE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PWM6_INT_UNDERFLOW_EN	PWM6_INT_FINISH_EN	PWM5_INT_UNDERFLOW_EN	PWM5_INT_FINISH_EN	PWM4_INT_UNDERFLOW_EN	PWM4_INT_FINISH_EN	PWM3_INT_UNDERFLOW_EN	PWM3_INT_FINISH_EN	PWM2_INT_UNDERFLOW_EN	PWM2_INT_FINISH_EN	PWM1_INT_UNDERFLOW_EN	PWM1_INT_FINISH_EN	PWM0_INT_UNDERFLOW_EN	PWM0_INT_FINISH_EN
Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM0_INT_FINISH_EN Set to 1 to enable PWM0 finish interrupt
PWM0_INT_UNDERFLOW_EN Set to 1 to enable PWM0 underflow interrupt
PWM1_INT_FINISH_EN Set to 1 to enable PWM1 finish interrupt
PWM1_INT_UNDERFLOW_EN Set to 1 to enable PWM1 underflow interrupt
PWM2_INT_FINISH_EN Set to 1 to enable PWM2 finish interrupt
PWM2_INT_UNDERFLOW_EN Set to 1 to enable PWM2 underflow interrupt
PWM3_INT_FINISH_EN Set to 1 to enable PWM3 finish interrupt
PWM3_INT_UNDERFLOW_EN Set to 1 to enable PWM3 underflow interrupt
PWM4_INT_FINISH_EN Set to 1 to enable PWM4 finish interrupt
PWM4_INT_UNDERFLOW_EN Set to 1 to enable PWM4 underflow interrupt
PWM5_INT_FINISH_EN Set to 1 to enable PWM5 finish interrupt
PWM5_INT_UNDERFLOW_EN Set to 1 to enable PWM5 underflow interrupt
PWM6_INT_FINISH_EN Set to 1 to enable PWM6 finish interrupt
PWM6_INT_UNDERFLOW_EN Set to 1 to enable PWM6 underflow interrupt

Note: Interrupt can not be supported when 32kHz clock. (OLD_MODE = 1, CLKSEL = 1)

PWM+0194h PWM Interrupt Status register PWM_INT_STATUS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PWM6_INT_UNDERFLOW_ST	PWM6_INT_FINISH_ST	PWM5_INT_UNDERFLOW_ST	PWM5_INT_FINISH_ST	PWM4_INT_UNDERFLOW_ST	PWM4_INT_FINISH_ST	PWM3_INT_UNDERFLOW_ST	PWM3_INT_FINISH_ST	PWM2_INT_UNDERFLOW_ST	PWM2_INT_FINISH_ST	PWM1_INT_UNDERFLOW_ST	PWM1_INT_FINISH_ST	PWM0_INT_UNDERFLOW_ST	PWM0_INT_FINISH_ST
Type			R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM0_INT_FINISH_ST PWM0 finish status
PWM0_INT_UNDERFLOW_ST PWM0 underflow status
PWM1_INT_FINISH_ST PWM1 finish status

PWM1_INT_UNDERFLOW_ST PWM1 underflow status
PWM2_INT_FINISH_ST PWM2 finish status
PWM2_INT_UNDERFLOW_ST PWM2 underflow status
PWM3_INT_FINISH_ST PWM3 finish status
PWM3_INT_UNDERFLOW_ST PWM3 underflow status
PWM4_INT_FINISH_ST PWM4 finish status
PWM4_INT_UNDERFLOW_ST PWM4 underflow status
PWM5_INT_FINISH_ST PWM5 finish status
PWM5_INT_UNDERFLOW_ST PWM5 underflow status
PWM6_INT_FINISH_ST PWM6 finish status
PWM6_INT_UNDERFLOW_ST PWM6 underflow status
Note: The interrupt status will be auto-cleared if interrupt enable or PWM enable is cleared.

PWM+0198h PWM Interrupt Acknowledge register PWM_INT_ACK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			PWM6_INT_UNDERFLOW_ACK	PWM6_INT_FINISH_ACK	PWM5_INT_UNDERFLOW_ACK	PWM5_INT_FINISH_ACK	PWM4_INT_UNDERFLOW_ACK	PWM4_INT_FINISH_ACK	PWM3_INT_UNDERFLOW_ACK	PWM3_INT_FINISH_ACK	PWM2_INT_UNDERFLOW_ACK	PWM2_INT_FINISH_ACK	PWM1_INT_UNDERFLOW_ACK	PWM1_INT_FINISH_ACK	PWM0_INT_UNDERFLOW_ACK	PWM0_INT_FINISH_ACK
Type			W	W	W	W	W	W	W	W	W	W	W	W	W	W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM0_INT_FINISH_ACK Set to 1 to clear PWM0 finish interrupt
PWM0_INT_UNDERFLOW_ACK Set to 1 to clear PWM0 underflow interrupt
PWM1_INT_FINISH_ACK Set to 1 to clear PWM1 finish interrupt
PWM1_INT_UNDERFLOW_ACK Set to 1 to clear PWM1 underflow interrupt
PWM2_INT_FINISH_ACK Set to 1 to clear PWM2 finish interrupt
PWM2_INT_UNDERFLOW_ACK Set to 1 to clear PWM2 underflow interrupt
PWM3_INT_FINISH_ACK Set to 1 to clear PWM3 finish interrupt
PWM3_INT_UNDERFLOW_ACK Set to 1 to clear PWM3 underflow interrupt
PWM4_INT_FINISH_ACK Set to 1 to clear PWM4 finish interrupt
PWM4_INT_UNDERFLOW_ACK Set to 1 to clear PWM4 underflow interrupt
PWM5_INT_FINISH_ACK Set to 1 to clear PWM5 finish interrupt
PWM5_INT_UNDERFLOW_ACK Set to 1 to clear PWM5 underflow interrupt
PWM6_INT_FINISH_ACK Set to 1 to clear PWM6 finish interrupt
PWM6_INT_UNDERFLOW_ACK Set to 1 to clear PWM6 underflow interrupt

PWM+01A0h PWM0 Control register PWM0_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS[5:0]						GUARD_VALUE	IDLE_VALUE	MODE	SRCSEL		CLKSEL	CLKDIV [2:0]		
Type	R/W	R/W						R/W	R/W	R/W	R/W		R/W	R/W		
Reset	0	3Fh						0	0	0	0		0	0		

CLKDIV Select PWM0 clock scale.

- **000** CLK Hz
- **001** CLK/2 Hz
- **010** CLK/4 Hz
- **011** CLK/8 Hz
- **100** CLK/16 Hz
- **101** CLK/32 Hz
- **110** CLK/64 Hz
- **111** CLK/128 Hz

CLKSEL Select PWM0 clock

- **0** CLK=CLKSRC
- **1** CLK=CLKSRC/1625

SRCSEL Select PWM0 data source

- **0** FIFO mode
- **2** Memory mode

MODE Select Random Generator mode

- 2 Periodical PWM mode.
- 3 Random PWM mode

Note: When using random generator mode, the data source comes from dual buffers in memory.

IDLE_VALUE PWM0 output value when idle state.

GUARD_VALUE PWM0 output value when guard time.

STOP_BITPOS The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.

OLD_PWM_MODE Use old PWM mode

- 2 New PWM mode
- 3 Old PWM mode

Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source could work in the system sleep-mode.

PWM+01A4h PWM0 High Duration register PWM0_HDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION[15:0]															
Type	R/W															
Reset	1															

HDURATION PWM0 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+01A8h PWM0 Low Duration register PWM0_LDURATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LDURATION[15:0]**															
Type	R/W															
Reset	1															

LDURATION PWM0 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.

Note: The duration of PWM must not be 0.

PWM+01ACh PWM0 Guard Duration register PWM0_GDRUATION

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GUARD_DURATION[15:0]**															
Type	R/W															
Reset	0															

GUARD_DURATION It's the guarding interval between individual waveforms and the output is decided by GUARD_VALUE. Also if it equals to N, it needs to program N-1 in this register.

Note: If this duration is 0, it means no guarding interval.

PWM+01B0h PWM0 Buffer0 Base Address register PWM0_BUF0_BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BUF0_BS_ADDR[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BUF0_BS_ADDR[15:0]**															
Type	R/W															
Reset	0															

BUF0_BS_ADDR The base address of memory buffer0 for PWM0's waveform data.

 © 2009 MediaTek Inc.

PWM+01B4h PWM0 Buffer0 Size register PWM0_BUF0_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF0_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF0_SIZE The length of the waveform data in memory buffer0 that PWM0 should generate. If it equals to N, need to program N-1 in this register.

Note: The size is in unit of 32-bit data.

PWM+01B8h PWM0 Buffer1 Base Address register PWM0_BUF1__BASE_ ADDR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUF1_BS_ADDR[31:16]															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_BS_ADDR[15:0]															
Type	R/W															
Reset	0															

BUF1_BS_ADDR The base address of memory buffer1 for PWM0's waveform data.

Note: The memory buffer1 is useless in periodical mode.

PWM+01BCh PWM0 Buffer1 Size register PWM0_BUF1_SIZE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUF1_SIZE[15:0]															
Type	R/W															
Reset	0															

BUF1_SIZE The length of the waveform data in memory buffer1 that PWM0 should generate. If it equals to N, need to program N-1 in this register.

PWM+01C0h PWM0 Send Data0 register PWM0_SEND_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0 [31:16]															
Type	R/W															
Reset	0															

 © 2009 MediaTek Inc.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA0[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA0 PWM0 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+01C4h PWM0 Send Data1 register PWM0_SEND_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA1[31:16]**															
Type	R/W															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA1[15:0]**															
Type	R/W															
Reset	0															

SEND_DATA1 PWM0 local buffer0 of pulse sequence data to be generated.

Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

PWM+01C8h PWM0 Wave Number register PWM0_WAVE_NUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAVE_NUM[15:0]**															
Type	R/W															
Reset	0															

WAVE_NUM The number by which PWM0 will generate from the pulse data repeatedly.

Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

PWM+01CCh PWM0 Data Width register PWM0_DATA_WIDTH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**DATA_WIDTH[12:0]**												
Type				R/W												
Reset				0												

DATA_WIDTH The PWM0 pulse data width in the old PWM mode.

PWM+01D0h PWM0 Thresh register PWM0_THRESH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**THRESH[12:0]**												
Type				R/W												
Reset				0												

THRESH The PWM0 pulse data high/low switching threshold in the old PWM mode.

PWM+01D4h PWM0 Send Wave Number register PWM0_SEND_WAVENUM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_WAVENUM[15:0]**															
Type	RO															
Reset	0															

SEND_WAVENUM The number by which PWM0 has already generated from the specified data source in the periodical mode.

PWM+01D8h PWM0 Valid register PWM0_VALID

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**BUF1_VALID_WEN**	**BUF1_VALID**	**BUF0_VALID_WEN**	**BUF0_VALID**
Type													W	R/W	W	R/W
Reset													0	0	0	0

BUF0_VALID The valid status is used to indicate pulse data in memory buffer0 is ready.
BUF0_VALID_WEN This bit must be set to modify BUF0_VALID.
BUF1_VALID The valid status is used to indicate pulse data in memory buffer1 is ready.
BUF1_VALID_WEN This bit must be set to modify BUF1_VALID.
Note: The program should set these bits after data are prepared in memory. The HW will clear these bits after it has used all data in the specified memory.

6.4 Real Time Clock

6.4.1 General Description

The Real Time Clock (RTC) module provides time and data information. The clock is based on a 32.768KHz oscillator with an independent power supply. When the mobile handset is powered off, a dedicated regulator supplies the RTC block. If the main battery is not present, a backup supply such as a small mercury cell battery or a large capacitor is used. In addition to providing timing data, an alarm interrupt is generated and can be used to power up the baseband core via the BBWAKEUP pin. Regulator interrupts corresponding to seconds, minutes, hours and days can be generated whenever the time counter value reaches a maximum value (e.g., 59 for seconds and minutes, 23 for hours, etc.). The year span is supported up to 2127. The maximum day-of-month values, which depend on the leap year condition, are stored in the RTC block.

6.4.2 Register Definitions

REGISTER ADDRESS	REGISTER NAME	SYNONYM
RTC+0000h	Baseband power up	RTC_BBPU
RTC+0004h	RTC IRQ status	RTC_IRQ_STA
RTC+0008h	RTC IRQ enable	RTC_IRQ_EN
RTC+000Ch	Counter increment IRQ enable	RTC_CII_EN
RTC+0010h	RTC alarm mask	RTC_AL_MASK
RTC+0014h	RTC seconds time counter register	RTC_TC_SEC
RTC+0018h	RTC minutes time counter register	RTC_TC_MIN
RTC+001Ch	RTC hours time counter register	RTC_TC_HOU
RTC+0020h	RTC day-of-month time counter register	RTC_TC_DOM
RTC+0024h	RTC day-of-week time counter register	RTC_TC_DOW
RTC+0028h	RTC month time counter register	RTC_TC_MTH
RTC+002Ch	RTC year time counter register	RTC_TC_YEA
RTC+0030h	RTC second alarm setting register	RTC_AL_SEC
RTC+0034h	RTC minute alarm setting register	RTC_AL_MIN
RTC+0038h	RTC hour alarm setting register	RTC_AL_HOU
RTC+003Ch	RTC day-of-month alarm setting register	RTC_AL_DOM
RTC+0040h	RTC day-of-week alarm setting register	RTC_AL_DOW
RTC+0044h	RTC month alarm setting register	RTC_AL_MTH
RTC+0048h	RTC year alarm setting register	RTC_AL_YEA
RTC+004Ch	XOSC bias current control register	RTC_XOSCCALI
RTC+0050h	RTC_POWERKEY1 register	RTC_POWERKEY1
RTC+0054h	RTC_POWERKEY2 register	RTC_POWERKEY2
RTC+0058h	PDN1	RTC_PDN1
RTC+005Ch	PDN2	RTC_PDN2
RTC+0064h	Spare register for specific purpose	RTC_SPAR1

 © 2009 MediaTek Inc.

RTC+0068h	Lock / unlock scheme to prevent RTC miswriting	RTC_PROT
RTC+006ch	One-time calibration offset	RTC_DIFF
RTC+0074h	Enable the transfers from core to RTC in the queue	RTC_WRTGR

Table 157 RTC Register Map

RTC+0000h Baseband power up RTC_BBPU

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY_BBPU								DBING	CBUSY	RELOAD	CLRPKY	AUTO	BBPU	WRITE_EN	PWREN
Type	WO								RO	RO	WO	WO	R/W	R/W	R/W	R/W

KEY_BBPU A bus write is acceptable only when KEY_BBPU=0x43.

DBING This bit indicates RTC is still de-bouncing.

CBUSY The read/write channels between RTC / Core is busy. This bit indicates high after software program sequence to anyone of RTC data registers and enable the transfer by RTC_WRTGR=1. By the way, it is high after the reset from low to high because RTC reload process. Notice: the CBUSY is always high in lock mode (powerkeys not match), please refer to the timeout period in RTC SOP documents.

RELOAD Reload the values from RTC domain to Core domain. Generally speaking, RTC will reload synchronize the data from RTC to core when reset from 0 to 1. This bit can be treated as debug bit.

CLRPKY Clear powerkey1 and powerkey2 at the same time. In some cases, software may clear powerkey1 & powerkey2. The BBWAKEUP depends on the matching specific patterns of powerkey1 and powerkey2. If any one of powerkey1 or powerkey2 or BBPU is cleared, BBWAKEUP goes low immediately. Software can't program the other control bits without power. By program RTC_BBPU with CLRPKY=1 and BBPU=0 condition, RTC can clear powerkey1, powerkey2 and BBPU at the same moment.

AUTO Controls if BBWAKEUP is automatically in the low state when SYSRST# transitions from high to low.

- **0** BBWAKEUP is not automatically in the low state when SYSRST# transitions from high to low.
- **1** BBWAKEUP is automatically in the low state when SYSRST# transitions from high to low.

BBPU Controls the power of PMIC. If powerkey1=A357h and powerkey2=67D2h, PMIC takes on the value programmed by software; otherwise PMIC is low.

- **0** Power down
- **1** Power on

WRITE_EN When WRITE_EN is write 0 by the MCU, the RTC programing interface is disabled immediately (MCU can't program RTC). After the debounce counter is time-out, the interface enabled again (MCU can program RTC). The debounce counter time-out period is decided by RTC_PDN1. Note that the WRITE_EN value read out is meaningless. The hardware only care about the "write-0 action" to WRITE_EN control bit.

When WRITE_EN==0, avoid to "read out RTC_BBPU, AND/OR something and write back", like this -> *RTC_BBPU=*RTC_BBPU|RTC_BBPU_KEY|0x1. This would disable RTC write interface for a while and hard to debug.

PWREN

- **0** RTC alarm has no action on power switch.
- 1 When an RTC alarm occurs, BBPU is set to 1 and the system powers on by RTC alarm wakeup.

RTC+0004h RTC IRQ status RTC_IRQ_STA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TCSTA	ALSTA
Type															R/C	R/C

ALSTA This register indicates the IRQ status and whether or not the alarm condition has been met.

- **0** No IRQ occurred; the alarm condition has not been met.
- **1** IRQ occurred; the alarm condition has been met.

TCSTA This register indicates the IRQ status and whether or not the tick condition has been met.

- **0** No IRQ occurred; the tick condition has not been met.
- **1** IRQ occurred; the tick condition has been met.

RTC+0008h RTC IRQ enable RTC_IRQ_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														ONESHOT	TC_EN	AL_EN
Type														R/W	R/W	R/W

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

ONESHOT Controls automatic reset of AL_EN and TC_EN.

AL_EN This register enables the control bit for IRQ generation if the alarm condition has been met.

- **0** Disable IRQ generations.
- **1** Enable the alarm time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.

TC_EN This register enables the control bit for IRQ generation if the tick condition has been met.

- **0** Disable IRQ generations.
- 1 Enable the tick time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.

RTC+000Ch Counter increment IRQ enable RTC_CII_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							1/8SECCII	1/4SECCII	1/2SECCII	YEACII	MTHCII	DOWCII	DOMCII	HOUCII	MINCII	SECCII
Type							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

SECCII Set this bit to 1 to activate the IRQ at each second update.

MINCII Set the bit to 1 to activate the IRQ at each minute update.

HOUCII Set the bit to 1 to activate the IRQ at each hour update.

DOMCII Set the bit to 1 to activate the IRQ at each day-of-month update.

DOWCII Set the bit to 1 to activate the IRQ at each day-of-week update.

MTHCII Set the bit to 1 to activate the IRQ at each month update.

YEACII Set the bit to 1 to activate the IRQ at each year update.

1/2SECCII Set the bit to 1 to activate the IRQ at each one-half of a second update.

1/4SECCII Set the bit to 1 to activate the IRQ at each one-fourth of a second update.

1/8SECCII Set the bit to 1 to activate the IRQ at each one-eighth of a second update.

RTC+0010h RTC alarm mask RTC_AL_MASK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										YEA_MSK	MTH_MSK	DOW_MSK	DOM_MSK	HOU_MSK	MIN_MSK	SEC_MSK
Type										R/W	R/W	R/W	R/W	R/W	R/W	R/W

The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Warning: If you set all bits 1 in RTC_AL_MASK (i.e. RTC_AL_MASK=0x7f) and PWREN=1 in RTC_BBPU, it means alarm comes EVERY SECOND, not disabled.

SEC_MSK

- **0** Condition (RTC_TC_SEC = RTC_AL_SEC) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_SEC = RTC_AL_SEC) is masked, i.e. the value of RTC_TC_SEC does not affect the alarm IRQ generation.

MIN_MSK

- **0** Condition (RTC_TC_MIN = RTC_AL_MIN) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_MIN = RTC_AL_MIN) is masked, i.e. the value of RTC_TC_MIN does not affect the alarm IRQ generation.

HOU_MSK

- **0** Condition (RTC_TC_HOU = RTC_AL_HOU) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_HOU = RTC_AL_HOU) is masked, i.e. the value of RTC_TC_HOU does not affect the alarm IRQ generation.

DOM_MSK

- **0** Condition (RTC_TC_DOM = RTC_AL_DOM) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_DOM = RTC_AL_DOM) is masked, i.e. the value of RTC_TC_DOM does not affect the alarm IRQ generation.

DOW_MSK

- **0** Condition (RTC_TC_DOW = RTC_AL_DOW) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_DOW = RTC_AL_DOW) is masked, i.e. the value of RTC_TC_DOW does not affect the alarm IRQ generation.

MTH_MSK

- **0** Condition (RTC_TC_MTH = RTC_AL_MTH) is checked to generate the alarm signal.
- **1** Condition (RTC_TC_MTH = RTC_AL_MTH) is masked, i.e. the value of RTC_TC_MTH does not affect the alarm IRQ generation.

YEA_MSK

- **0** Condition (RTC_TC_YEA = RTC_AL_YEA) is checked to generate the alarm signal.
- 1 Condition (RTC_TC_YEA = RTC_AL_YEA) is masked, i.e. the value of RTC_TC_YEA does not affect the alarm IRQ generation.

RTC+0014h RTC seconds time counter register RTC_TC_SEC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											TC_SECOND					
Type											R/W					

TC_SECOND The second initial value for the time counter. The range of its value is: 0-59.

RTC+0018h RTC minutes time counter register RTC_TC_MIN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**TC_MINUTE**					
Type											R/W					

TC_MINUTE The minute initial value for the time counter. The range of its value is: 0-59.

RTC+001Ch RTC hours time counter register RTC_TC_HOU

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**TC_HOUR**				
Type												R/W				

TC_HOUR The hour initial value for the time counter. The range of its value is: 0-23.

RTC+0020h RTC day-of-month time counter register RTC_TC_DOM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**TC_DOM**				
Type												R/W				

TC_DOM The day-of-month initial value for the time counter. The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

RTC+0024h RTC day-of-week time counter register RTC_TC_DOW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**TC_DOW**		
Type														R/W		

TC_DOW The day-of-week initial value for the time counter. The range of its value is: 1-7.

RTC+0028h RTC month time counter register RTC_TC_MTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**TC_MONTH**			
Type													R/W			

TC_MONTH The month initial value for the time counter. The range of its value is: 1-12.

RTC+002Ch RTC year time counter register RTC_TC_YEA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										**AL_SECOND**						
Type										R/W						

TC_YEAR The year initial value for the time counter. The range of its value is: 0-127. (2000-2127)

RTC+0030h RTC second alarm setting register RTC_AL_SEC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**AL_SECOND**					
Type											R/W					

AL_SECOND The second value of the alarm counter setting. The range of its value is: 0-59.

© 2009 MediaTek Inc.

RTC+0034h RTC minute alarm setting register RTC_AL_MIN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											AL_MINUTE					
Type											R/W					

AL_MINUTE The minute value of the alarm counter setting. The range of its value is: 0-59.

RTC+0038h RTC hour alarm setting register RTC_AL_HOU

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												AL_HOUR				
Type												R/W				

AL_HOUR The hour value of the alarm counter setting. The range of its value is: 0-23.

RTC+003Ch RTC day-of-month alarm setting register RTC_AL_DOM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												AL_DOM				
Type												R/W				

AL_DOM The day-of-month value of the alarm counter setting. The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

RTC+0040h RTC day-of-week alarm setting register RTC_AL_DOW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														AL_DOW		
Type														R/W		

AL_DOW The day-of-week value of the alarm counter setting. The range of its value is: 1-7.

RTC+0044h RTC month alarm setting register RTC_AL_MTH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													AL_MONTH			
Type													R/W			

AL_MONTH The month value of the alarm counter setting. The range of its value is: 1-12.

RTC+0048h RTC year alarm setting register RTC_AL_YEA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										AL_YEAR						
Type										R/W						

AL_YEAR The year value of the alarm counter setting. The range of its value is: 0-127. (2000-2127)

RTC+004Ch XOSC bias current control register RTC_XOSCCALI

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												XOSCCALI				
Type												WO				

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

© 2009 MediaTek Inc.

XOSCCALI This register controls the XOSC32 bias current.

RTC+0050h RTC_POWERKEY1 register RTC_POWERKEY1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RTC_POWERKEY1															
Type	R/W															

RTC+0054h RTC_POWERKEY2 register RTC_POWERKEY2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RTC_POWERKEY2															
Type	R/W															

These register sets are used to determine if the real time clock has been programmed by software; i.e. the time value in real time clock is correct. When the real time clock is first powered on, the register contents are all undefined, therefore the time values shown are incorrect. Software needs to know if the real time clock has been programmed. Hence, these two registers are defined to solve this power-on issue. After software programs the correct value, these two register sets do not need to be updated. In addition to programming the correct time value, when the contents of these register sets are wrong, the interrupt is not generated. Therefore, the real time clock does not generate the interrupts before the software programs the registers; unwanted interrupt due to wrong time value do not occur. The correct values of these two register sets are:

RTC_POWERKEY1 A357h
RTC_POWERKEY2 67D2h

RTC+0058h PDN1 RTC_PDN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RTC_PDN1[7:0]							
Type									R/W							

RTC_PDN1[3:1] is for reset de-bounce mechanism. When RTC_POWERKEY1 & RTC_POWERKEY2 do not match the correct values, RTC_PDN1[3:1] is set to 3(011 in binary).

- **0** 2ms
- **1** 8ms
- **2** 32ms
- **3** 128ms
- **4** 256ms
- **5** 512ms
- **6** 1024ms
- **7** 2048ms

RTC_PDN1[7:4] & RTC_PDN1[0] is the spare register for software to keep power on and power off state information.

RTC+005Ch PDN2 RTC_PDN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									**RTC_PDN2[7:0]**							
Type									R/W							

RTC_PDN2 The spare register for software to keep power on and power off state information.

RTC+0064h Spare register for specific purpose RTC_SPAR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RTC_SPAR1**															
Type	R/W															

RTC_SPAR1 These registers are reserved for specific purpose.

RTC+0068h Lock / unlock scheme to prevent RTC miswriting RTC_PROT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RTC_PROT**															
Type	R/W															

RTC_PROT The RTC write interface is protected by RTC_PROT. Whether the RTC writing interface is enabled or not is decided by RTC_PROT contents. When RTC_POWERKEY1 & RTC_POWERKEY2 are not equal to the correct values, the RTC writing interface is always enabled. But when they match, users have to perform Unlock flow to enable the writing interface.

Unlock flow: Step1: *RTC_PROT=0x9136; // compare 15 bits: bit 15~1
Step2: *RTC_WRTRG=1;
Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec

Lock flow: Step1: *RTC_PROT=0x0;
Step2: *RTC_WRTRG=1;
Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec

Once the normal RTC content writing is complete, it is suggested to perform Lock flow to turn off the interface to avoid accident writing.

RTC+006ch One-time calibration offset RTC_DIFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**RTC_DIFF**											
Type					R/W											

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

RTC_DIFF These registers are used to adjust the internal counter of RTC. It effects once and returns to zero in done.

In some cases, you observe the RTC is faster or slower than the standard. To change RTC_TC_SEC is coarse and may cause alarm problem. RTC_DIFF provides a finer time unit. An internal 15-bit counter accumulates in each 32768-HZ clock. Entering a non-zero value into the RTC_DIFF causes the internal RTC counter increases or decreases RTC_DIFF when RTC_DIFF changes to zero again. RTC_DIFF represents as 2's completement form.

For example, if you fill in 0xfff into RTC_DIFF, the internal counter decreases 1 when RTC_DIFF returns to zero. In other words, you can only use RTC_DIFF continuously if RTC_DIFF is equal to

© 2009 MediaTek Inc.

zero now.

Note: RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045). 0x7ff & 0x7fe are forbid to use.

RTC+0074h Enable the transfers from core to RTC in the queue RTC_WRTGR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**WRTGR**
Type																WO

WRTGR This register enables the transfers from core to RTC. After you modify all the RTC registers you'd like to change, you must write RTC_WRTGR to 1 to trigger the transfer. The prior writing operations are queued at core power domain. The pending data will not be transferred to RTC domain until WRTGR=1.
After WRTGR=1, the pending data is transferred to RTC domain sequentially in order of register address, from low to high. For example: RTC_BBPU -> RTC_IRQ_EN -> RTC_CII_EN -> RTC_AL_MASK -> RTC_TC_SEC -> etc. The CBUSY in RTC_BBPU is equal to 1 in writing process. You can observe CBUSY to determine when the transmission completes.